ALEKSANDAR DIMITROV SAVKOV

DECIPHERING CLINICAL TEXT

DECIPHERING CLINICAL TEXT

Concept Recognition in Primary Care Text Notes

ALEKSANDAR DIMITROV SAVKOV

Philosophiae Doctor (PhD)

Department of Informatics

School of Engineering and Informatics

University of Sussex

December 2015

DECLARATION

This thesis, whether in the same or different form, has not been previously submitted to this or any other University for a degree. Some of the ideas and figures in this thesis have been previously published as noted below. The work on designing a set of annotation guidelines described in Chapter 3 was jointly produced with Prof. John Carroll, Prof. Jackie Cassell, and Dr. Rob Koeling. The Harvey corpus annotation with chunks and medical concepts was carried out by Mwenya Kosomo, Katie Ellis, Lauren Bignell, Matthew Cadd, and Fiona Dogan under my supervision.

Brighton, UK, December 2015

Aleksandar Dimitrov Savkov

Dedicated to the loving memory of Dechko Yordanov Ivanov

1929 – 2014

To mum and dad

The most exciting phrase to hear in science,
the one that heralds new discoveries, is not "Eureka!",
but "That's funny..."

— Isaac Asimov

ABSTRACT

Electronic patient records, containing data about the health and care of a patient, are a valuable source of information for longitudinal clinical studies. The General Practice Research Database (GPRD) has collected patient records from UK primary care practices since the late 1980s. These records contain both structured data (in the form of codes and numeric values) and free text notes. While the structured data have been used extensively in clinical studies, there are significant practical obstacles in extracting information from the free text notes. The main obstacles are data access restrictions, due to the presence of sensitive information, and the specific language of medical practitioners, which renders standard language processing tools ineffective.

The aim of this research is to investigate approaches for computer analysis of free text notes. The research involved designing a primary care text corpus (the Harvey Corpus) annotated with syntactic chunks and clinically-relevant semantic entities, developing a statistical chunking model, and devising a novel method for applying machine learning for entity recognition based on chunk annotation. The tools produced would facilitate reliable information extraction from primary care patient records, needed for the development of clinically-related research. The three medical concept types targeted in this thesis could contribute to epidemiological studies by enhancing the detection of co-morbidities, and better analysing the descriptions of patient experiences and treatments.

The main contributions of the research reported in this thesis are: guidelines for chunk and concept annotation of clinical text, an approach to maximising agreement between human annotators, the Harvey Corpus, a method for using a standard part-of-speech tagging model in clinical text chunking, and a novel approach to recognising clinically-relevant medical concepts.

PUBLICATIONS

Some ideas and figures in this thesis have appeared previously in the following publications:

Savkov, A., Carroll, J., and Cassell, J. (2014). Chunking Clinical Text Containing Non-canonical Language. In *Proceedings of the 13th BioNLP Workshop*, Baltimore, USA.

Savkov, A., Carroll, J., Koling, R., and Cassell, J. (in print 2015). Annotating Patient Clinical Records with Syntactic Chunks and Named Entities. *Language Reources and Evaluation*.

ACKNOWLEDGMENTS

A friend of mine once said that despite its purpose of producing a new independent researcher, a doctorate is in fact a team effort — one led by the student, but contributed to by his or her supervisors and a whole department or school.

I should thank first and foremost my supervisors *Prof. John Carroll* and *Prof. Jackie Cassell*, without whose advice, attention, and collaboration producing this thesis would have been infinitely more difficult. Thank you for your support!

I am incredibly grateful to my good friends and colleagues *Matti Lyra* and *Miroslav Batchkarov*, with whom I shared an office for the longest time, for being a motivating benchmark, as well as the occasional rubber duck or an emotional punching bag. I should also mention other members of the department and the Brighton and Sussex Medical School *Dr. Novi Quadrianto*, *Dr. Rob Koeling*, and *Dr. Elizabeth Ford*, who contributed to my research with ideas and advice.

A special thank you and a hat tip to *Jeremy Maris* and *Christian Caterham* for the work they did in supporting the staff and students at the department.

Special thanks to my friends *Dr. Kevin B. Cohen* and *Dr. Kostadin Cholakov* who provided me with critical advice on several occasions.

My whole research is underpinned by a dataset that was annotated by a group of medical students who took interest in helping me and made time in their busy schedules to listen to me blabber about language, linguistics, and other seemingly boring subjects. My sincere thanks to *Mwenya Kosomo*, *Lauren Bignell*, *Katie Ellis*, *Mattew Cadd*, and *Fiona Dogan*.

Also a very special thank you to Thomas Kober for providing high quality proof reading on a short notice. I hope I can return the favour!

Finally, it is difficult to express how much my family has helped me through the past years, silently caring, worrying, and supporting me from afar. I am truly grateful for all of it!

Thank you!

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AI	Artificial Intelligence
AIDS	Acquired Immune Deficiency Syndrome
AP	Adjectival Phrase
BEISO	Beginning, End, Inside, Single, Outside
BIO	Beginning, Inside, Outside
BNC	British National Corpus
BO	Bayesian Optimisation
BoW	Bag-of-Words
CL	Computational Linguistics
CRAFT	Colorado Richly Annotated Full-Text Corpus
CRF	Conditional Random Field
CS	Computer Science
CTV3	Clinical Terminology Version 3
CV	Cross-Validation
DC	Divide & Conquer
EMR	Electronic Medical Records
GP	General Practitioner
GPRD	General Practice Research Database
HIPAA	Health Insurance Portability and Accountability Act

HIV Human Immunodeficiency Virus

HLBL Hierarchical Log-Bilinear (Embeddings)

HPSG Head-Driven Structure Grammar

IAA Inter-Annotator Agreement

IE Information Extraction

IR Information Retrieval

kNN k-Nearest Neighbours

KPSC Kaiser Permanente Southern California

LE Locative Expression

LFG Lexical Function Grammar

MaxEnt Maximum Entropy

MHRA Medicines and Healthcare Products Regulatory Agency

ML Machine Learning

MUC-7 Seventh Message Understanding Conference

MV Main Verb

NB Naïve Bayes

NE Named Entities

NER Named Entity Recognition

NLP Natural Language Processing

NHS National Health Service

NP Noun Phrase

OE On-Examination Expression

OXMIS Oxford Medical Information System

PCFG Probabilistic Context-Free Grammar

PHI Protected Health Information

POS Part of Speech

PP Prepositional Phrase

PPMI Positive Pointwise Mutual Information

PREP Patient Records Enhancement Programme

PTB Penn TreeBank

QE Quantitative Expression

RBF Radial Basis Function

SVM Support Vector Machine

TAG Tree-Adjoining Grammar

TE Temporal Expression

tf-idf term frequency - inverted document frequency

UMLS Unified Medical Language System

UPMC University of Pittsburgh Medical Center

UT Universal Tagset

WSD Word Sense Disambiguation

WSJ Wall Street Journal

NOTATION

The notation used in this thesis generally follows the conventions in NLP literature. The notation related to *inter-annotator agreement* is based on the one used by Artstein and Poesio (2008) with minor changes aimed at better consistency. Also the term *coder* is replaced with *annotator* in order to comply with the terminology in the rest of the thesis.

A	inter-annotator agreement (IAA)
A_o	observed IAA
A_e	IAA expected by chance
K	set of annotation categories
C	set of annotators
I	set of annotation items
n	number of annotation items $\|I\|$
η	number of judgment pairs $\binom{n}{2}$
m_i	number of annotators that have assigned a category to item i
S	IAA coefficient defined by Bennett et al. (1954)
π	IAA coefficient defined by Scott (1955)
κ	IAA coefficient defined by Cohen (1960)
α	IAA coefficient defined by Krippendorff (1980, 2004)
δ	difference function appropriate to the metric of the data; used for *Krippendorf's* α
X	matrix of annotation categories with annotation items as rows and annotators as columns; $X_{i,c}$ is the annotation category to which annotator c assigned item i

Y	matrix of annotation counts with annotation items as rows and annotation categories as columns; $Y_{i,k}$ is the number of annotators that assigned item i to category k
Z	coincidence matrix of annotation categories; $Z_{i,j}$ is the coincidence of the i-j category pair
Θ	square matrix of disagreement weights between categories; $\Theta_{i,j}$ is the disagreement weight of the category pair i and j; weights are in the range $[0, 1]$ zero being the agreement
$P(x)$	probability of x
$\hat{P}(x)$	observed probability of x

Upper case letters are used for sets and their lower case versions for the set members, e.g. k is a category from the set of annotation categories K. The size of sets is denoted by straight lines — the number of annotation categories is $|K|$.

1

INTRODUCTION

Clinical text — text written by healthcare workers about the care given to individual patients — is a source of rich, detailed information that could be of great use for health service planning and for the study of disease. However, unlocking that information at scale for research purposes is hindered by processing difficulties caused by the peculiarities of clinical language use, and the limited available development data due to the presence of sensitive information that could potentially identify patients. An important research goal is to achieve a reliable language processing foundation to allow more complex information extraction (IE) tasks to reach a sufficiently reliable performance level. Achieving this goal would allow the use of automated algorithms for processing clinical text in secure storage environments, thereby allowing researchers to analyse data without accessing it directly. Such analysis will avoid the manual de-identification currently required for researchers accessing the data, which will decrease the time and financial costs of their work.

Over the past fifteen years or so, Natural Language Processing (NLP) technology has reached a state of maturity that has allowed it to be used in a diverse range of real-world applications, some involving clinical text. Most NLP systems are developed on standard, grammatical, edited text, and are intended to be applied to text of the same type. Their accuracy is significantly degraded when applied to clinical text. Therefore, researchers have created new clinical text corpora to facilitate the development of accurate NLP tools in the clinical domain. Apart from terminology and some idiosyncratic expressions, discharge summaries — the predominant document type in clinical corpora — typically consist of well formed descriptive grammatical text. In contrast, progress notes and primary care notes give rise to difficult language processing issues arising from their typically heavy use of abbreviations, acronyms, medical jargon, ungrammatical constructions, and other

non-canonical (or non-standard) language. The goal of the studies conducted for this thesis is to achieve reliable automatic concept extraction for UK primary care text, adapting existing machine learning (ML) technology and language resources where appropriate, and developing new technology and research when necessary. The medical concepts targeted in the studies are *diseases*, *symptoms* and *signs* of diseases, and *drug names*. In the context of clinical notes, those are mentions of diseases and drug names, and records of symptoms stated by the patients and signs observed by the medical practitioner. On the one hand these concepts are interesting for epidemiological, pathological and pharmaceutical research, while on the other existing technology can be adapted to their natural language occurrence patterns.

While the trivial way to identify these concepts is to compile an approximation of a comprehensive keyword list, in reality such an approach has two significant limitations. First, keyword lists are not easily scalable. Compiling an exhaustive list of keywords for a single concept may seem like an easy enough task, but scaling the list to all possible concepts is an arduous task, especially in the clinical domain where the same concept may be expressed in many and unpredictable ways. Second, even a well crafted keyword list needs to deal with false positives — not all contexts of a word may align with the presumed meaning. On the other hand, recent advances in NLP technology could allow more accurate and scalable recognition of clinical concepts. Given the challenges of the language of primary care data, it is difficult develop solutions for all components of such a system to their respective state-of-the-art levels, but constructing a robust prototype could 1) serve as proof of concept for further research in exploring primary care data, and 2) confirm the feasibility of the approach.

1.1 DATA BACKGROUND

Large samples of primary care electronic patient records in the UK are collected and kept in regional and national databases, the largest of which is the General Practice Research Database (GPRD), hosted at the UK's Medicines and Healthcare Products Regulatory Agency (MHRA). These records have a structured part, based on the Read code system (Bentley et al., 1996), and a free-text part also referred to as a note. The primary care

data relevant for this thesis was manually de-identified and supplied to the Patient Records Enhancement Programme (PREP)[1] by the GPRD under a licence for research purposes.

1.1.1 *Read Code System*

The Read codes are a clinical terminology system used in primary care in the United Kingdom. The Read code system encodes a wide variety of patient data including symptoms and observations, diagnosis, performed procedures, gender, race, religion, and others, including administrative items such as missed appointments. The original codes in the system were developed in the 1980s by Dr. James Read to facilitate the input of certain aspects of patient care into a computer system. There are three designs of the code system, the latest of which, the Clinical Terminology Version 3 (CTV3), was mandated by the UK National Health Service (NHS) for standard use in general practice electronic medical records (EMR). In its latest designs the codes are structured as a polyhierarchy of indefinite depth, meaning that a code in the hierarchy can have more than one parent. The CTV3 system is set to be phased out of primary care, and replaced by its successor SNOMED-CT by the end of 2016[2].

1.1.2 *The General Practice Research Database*

The GPRD is a longitudinal database of primary care medical records. It contains comprehensive observational data from general practices, which makes it a valuable resource for a broad range of research areas, such as clinical epidemiology, disease patterns, disease management, research outcomes, and drug utilisation. Its data consists of primary care medical records; in these, general practitioners (GP) and other healthcare workers input information on events regarding their patients as structured data and free text. The structured data varies between the several software systems for primary care certified by the National Health Service, although a Read code and a term associated with it are always present in each record (see Figure 1.1).

1 http://prep.sussex.ac.uk/
2 Mapping tables between the standards can be found at https://isd.hscic.gov.uk/trud3/.

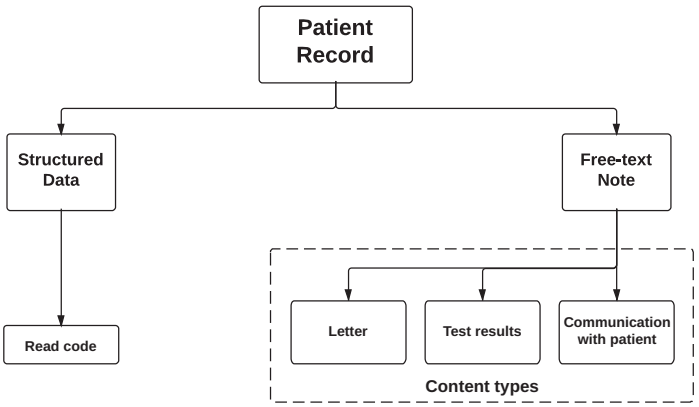


Figure 1.1: Patient record content diagram

The language and content of the free text is related to the role of GPs in the NHS. They are the gatekeepers to specialist care, charged with basic care for the patients, and with initial assessment and recommendation for specialist treatment. They are organised in practices of one or a small number of general practitioners, set up independently from the hospital system. Apart from correspondence with specialists, GP notes are mainly intended for use within the general practice in which they were created.

1.1.3 *Research Using UK Primary Care Data*

The information in UK primary care records is an important medical research resource, but so far only a small fraction of the information in the free text has been extracted and utilised. Some of the first studies in this area show that information gathered from free text notes has great potential.

The Freetext Matching Algorithm (Shah et al., 2012) is an automated method for extracting information from free text. The algorithm uses dictionaries of Read code terms (the text representation of Read codes) and "regular" words, as well as spelling correction software to make the language more canonical (i.e. more like standard English). Then it uses synonym look-up tables and phrase patterns to identify diagnoses, dates, and selected kinds of test results. The algorithm creates approximate matches between words and expressions in the free text on one side, and Read codes and OXMIS codes[3] on the other.

3 The Oxford Medical Information System (OXMIS) was an earlier terminology system used in primary care computer systems from 1987, when GPRD started. Practices switched over to the Read code system at different times in the 1990s (Shah et al., 2012).

It was tested on two sets of 1,000 records — one general and one associated with death — each taken from the GPRD. The algorithm achieved 98 precision and 93 recall on the death related dataset, and 92 precision and 77 recall on the other dataset. The authors also presented a cause of death detection algorithm aided by the Freetext Matching Algorithm to deal with records where the cause of death was recorded only in the free text. They conclude that the algorithm has achieved sufficient precision, and may facilitate research using patient record free text, particularly for extracting cause of death.

Koeling et al. (2011b) describe the development of a method for automatically determining common symptoms of ovarian cancer in free text notes. The model was based on 344 annotated records of women in the year prior to an ovarian cancer diagnosis. The study was concerned with finding the incidence of five common symptoms of ovarian cancer. Through manual annotation of notes, the estimates of the incidence of symptoms increased by 40% or more when the coded data was augmented with the free text annotation. The automatic symptom detection method was able to extract a significant proportion of this extra information (46 recall) with high precision (96). The automated approach developed for the study was intended to aid medical researchers wishing to validate studies based on codes, or to accurately assess symptoms, using information that can be automatically extracted from free text.

1.2 THE CHALLENGES

While great efforts are being made to process, interlink, and reuse the structured part of primary care patient records, as well as secondary care data (Frederick, 2003; Ávarez et al., 2011; Trust, 2015), very few studies have used the information in the free text notes. Details about symptoms and diseases typed by general practitioners, have not only the potential to enrich the majority of their structured information counterparts, but in many cases they can be the *only source of relevant information*. The latter is well illustrated by structured (coded) data entries such as `had a chat to patient` and `telephone encounter`, which have no association with a medical concept and rely solely on the information recorded in the associated text to convey details about the patient encounter (see Example 1.1). Similar arguments are made by Stubbs et al. (2015b), moti-

vating the organisation of the 2014 i2b2 track on recognising risk factors associated with heart disease.

Telephone encounter	tel from wife pt v scared re mri next wed- ok for small dose dz
Constipation NOS	1 BM 3 days ago following 5 days without any. now no BM last 3 days either. breast fed baby ! o/e abd soft. no palpable faeces. try lactulose 2.5 ml bd
Cardiac failure therapy	Hxnsyx settled ? feels abit better OE creps R base only. jvp not seen. IMP better re fluid status, rate still ok. P cont w bloods 2/7, rv 1w
Had a chat to patient	re. cough at night; see letter from Mr ~~~~~

Example 1.1: Examples of examination records from the GPRD consisting of a structured entry (left) and a text note (right).

It is important to compare the more refined language of the MED corpus of clinical notes (Coden et al., 2005), shown in Example 1.2, to the one demonstrated in Example 1.1. The differences are even starker when primary care text is compared to the terminology-rich biomedical text of GENIA (Ohta et al., 2002), or the edited news text of the Penn Treebank (Marcus et al., 1993).

A good idea of the importance of the information contained in primary care text can be drawn from a review of information extraction research focused on secondary care text, such as the i2b2 challenges (Uzuner et al., 2007a, 2010b, 2011; Sun et al., 2013a) and the Conference and Labs of the Evaluation Forum (CLEF) initiative (Roberts et al., 2009). Some of the studies focus on identifying specific *concepts* like a subset of symptoms, drugs and time expressions, while others also aim to recognise *relations* between them, e.g. the locus of a medical finding or the frequency of the administration of a drug. Other studies, such as the one reported by Uzuner (2009), look even further and consider relations between concepts spread throughout the medical history of a patient. Studies with such scope and the tools they produce approach the real-life needs for NLP tools in large scale clinical research. Aiming to achieve that is the long term goal of processing primary care text.

Although the coreference resolution study by Uzuner et al. (2012) showed good results for rule-based and hybrid methods, the majority of research into NLP technologies for

MED # 1 Left ACL disruption, return-to-work evaluation Patient of Dr. NAME. Samples mailed to home address. Patient is on Prilosec 20 mg bid. The ACE level remains in the lower limit of normal. Total cholesterol is 160 with an HDL cholesterol of 43, and LDL of 92, and a triglyceride of 123.

GENIA TI - IL-2 gene expression and NF-jB activation through CD28 requires reactive oxygen production by 5-lipoxygenase. AB-activation of the CD28 surface receptor provides a major costimulatory signal for T-cell activation resulting in enhanced production of interleukin-2 (IL-2) and cell proliferation.

PTB The asbestos fiber, crocidolite, is unusually resilient once it enters the lungs, with even brief exposures to it causing symptoms that show up decades later, researchers said. Lorillard Inc., the unit of New Yorkbased Loews that makes Kent cigarettes, stopped using crocidolite in its Micronite cigarette filters in 1956.

Example 1.2: Text samples from the MED (Coden et al., 2005), GENIA (Ohta et al., 2002), and Penn Treebank (Marcus et al., 1993) corpora provided by Coden et al. (2005).

processing secondary care text has been based on machine learning techniques. In contrast, the few studies that have successfully made use of UK primary care clinical notes mostly use heuristics and rule-based algorithms to achieve their goals (Koeling et al., 2011b; Shah et al., 2012). No significant studies to date have applied machine learning methods to them. The main reason is the difference between the non-canonical language of the notes, and the grammatically well-formed language of edited text that is normally targeted in NLP research. Primary care notes are characterised by extreme brevity of expression, numerous medical terms and jargon, ungrammatical constructions, spelling mistakes, and irregular and unorthodox usage of punctuation. Successful use of machine learning based methods benefits from an adequate choice of task and sufficient amount of training data. Therefore, a corpus of reasonable size needed to be developed. Certain similarities exist between primary care text notes and some types of secondary care text such as radiology reports and progress notes, but applying tools trained on the latter to primary care text would not produce optimal results. There are differences in the types of language they use, as well as in the topics they discuss – general practices use a much wider variety of topics and vocabulary compared to radiology reports for instance.

A key factor in the development of NLP tools for specific high level tasks is the reliability of core processing: text normalisation, part-of-speech (POS) tagging, chunking, and parsing. These processing steps supply an important part of the information needed for

good application performance. Syntactic chunks, for instance, are an important feature for concept recognition models. Using existing tools developed for other kinds of text would save development effort, but differences in the type of language and vocabulary might degrade their performance too much. Therefore, the first challenge is to find out which tools and resources can be used or adapted to primary care notes, and which ones should be rebuilt from scratch. Once a set of reliable core NLP processes is established, these can support the development of concept recognition models.

1.3 A PROCESSING ROAD MAP TO CONCEPT EXTRACTION

Extracting medical concepts, such as symptoms, diseases, and drug names, from clinical text is in essence similar to Named Entity Recognition (NER) in generic text. Both aim to identify various types of multi-token constructions representing some sort of semantic entity, and have been approached in similar ways in previous research involving clinical text (Uzuner, 2009; Uzuner et al., 2010a, 2011). A processing pipeline producing concept annotation is typically made up of a *tokeniser* and a *sentence splitter*, a *part-of-speech tagger*, a *chunker* or a *constituency parser*, and a *concept recognition module*. There are readily available tools and models to build a processing pipeline that prepares data for developing a concept recognition model, but the cost of applying this successfully across domains is unclear. Evaluating them in some way on primary care text was needed before drawing a clear road map for achieving reliable medical concept extraction. Ideally, such an evaluation would use a substantial amount of annotated gold standard data, but unfortunately such a resource was not available at the beginning of the work described in this thesis.

Another potential alternative is evaluating the tools on a small manually annotated data set, but the results of such an evaluation would most likely be misleading given the data sparsity. This approach would also not provide much insight into the shortcomings of the tools. Instead, an approach of observing and analysing the errors manually was used to determine the applicability of tools and statistical models. The following paragraphs describe the observations made on initial attempts to apply core NLP analysis tools to the data.

TOKENISATION AND SENTENCE SPLITTING are fundamental tasks in NLP applications, yet there is relatively little research in this area due to the widespread use of established gold standard annotated corpora containing segmented text (Dridan and Oepen, 2012). Although there are a number of segmentation tools developed for less canonical types of text (Kulick et al., 2004b; Gimpel et al., 2011), adapting one of them still left too many segmentation issues unresolved, and the errors would inevitably propagate in further processing. The observed tokenisation errors, although unusual, were simple, and could be addressed by developing a tokeniser specific to this type of text. In contrast, sentence structures in the primary care text could not be identified by the tools, nor by humans for that matter, due to missing sentence-level syntax in this telegraphic style of expression.

PART-OF-SPEECH TAGGING is the task of assigning parts of speech to words or tokens. The assignment is mostly dependent on the word itself and the immediately preceding and following words, so it should remain mostly unaffected by the terseness of primary care text. Unknown words were an expected source of errors, due to the lack of coverage of clinical vocabulary in the text the model was trained on (Penn Treebank described in Marcus et al. 1993 and GENIA described in Ohta et al. 2002). Despite the issues, the POS annotation seemed to be at a level of accuracy that would be sufficient for recognising higher level syntactic structure.

CHUNKING is preferred to other forms of syntactic parsing in NER-like tasks, because it focuses on phrases rather than sentences, thereby minimising the error rate while retaining the most important information about the syntactic structure. A few chunking tools (YAMCHA models as described in Kudo and Matsumoto 2003, CRF++ models[4], and GENIA as described in Kulick et al. 2004b) were applied in order to draw general conclusions about errors. Chunking used POS annotation generated by the Stanford POS tagger (Toutanova et al., 2003). The chunkers correctly identified most chunks, but often attempted building longer chunks and thereby wrongly attached part of a neighbouring chunk, or merged two smaller chunks into one. Other causes of errors were error propagation from the POS annotation, and unseen words. Despite the many mistakes, the obser-

4 CRF++ and the models are available at http://taku910.github.io/crfpp/

vations showed that if short (base) noun phrases could be recognised, chunking could be adapted to primary care text, given sufficient amount of annotated text from the domain. This observation was important, as it showed that chunking is a potentially successful approach to the syntactic analysis of this type of primary care text but only if the models are trained on annotated data from the same domain. The reason a new annotation was necessary was that all other available English language resources that involved chunking were produced by trimming down constituency parse trees.

FULL PARSING , being a more complex sentence-based syntactic analysis, had little chance of presenting a better alternative to chunking considering the qualities of the data; however, an overview of the errors seemed beneficial for the better understanding of the challenges presented by the data. The Stanford dependency parser (Manning, 2011) and the RASP constituency parser (Briscoe et al., 2006) were used for the tests. Due to the lack of sentence structure, the data was input to the parsers without any sentence segmentation. An important observation became obvious: the lack of sentence units caused the parsers to make major errors. They would try to stitch relatively well parsed small segments into a sentence, when in fact they were separate sentence-like units with no syntactic connection.

THE PLAN for the research described in this thesis took into account the preliminary observations and analysis outlined above. Achieving satisfactory medical concept recognition would require most NLP processes to be adapted to the target data. A new tokeniser was necessary to deal with the text peculiarities; an existing POS tagger could be used, but the most suitable one had to be determined; a corpus with chunk annotation was needed to develop an accurate chunker; finally, a concept annotation model could be built taking the NP chunk annotation as input.

THE MOTIVATION for this particular way of developing a system has several key aspects. Ideally, a new dataset with all core NLP annotations would be needed to address all of the issues identified above, but due to limited time and resources, a cheaper and faster compromise had to be reached. Noun chunking offered a good chance to eliminate the flaws of the readily available models, while relying on the relatively good performance of POS tagging models. It also provided a platform for concept annotation at a low cost, assuming

certain qualities of medical concepts. The assumption was that since the targeted concepts are noun-centric constructions — symptoms, drug names, diseases — noun chunks can be used as candidate units for annotation, i.e. after (noun) chunks are annotated, the output annotations are annotated again with concept types (or nothing). This style of annotation simplified the task, and consequently the preparation for it. These advantages of chunking made it the most promising annotation type out of the three in consideration.

AN ALTERNATIVE approach would be to use a parsing method designed to deal with ungrammatical sentences. The approach suggested by Foster (2007) allows parsing models to overcome grammar or spelling errors in a sentence. Fan et al. (2013) implements a method following this approach. It accounts for missing and spurious words by adding special nodes in the annotation. It also simplifies the internal structure of some of the phrases, making them flatter to avoid errors caused by unusual or ungrammatical expressions. This approach is more complex than chunking, yet only slightly more informative. Another major problem with applying it to the primary care data was the lack of identified sentence boundaries.

ANOTHER ALTERNATIVE approach would be using domain adaptation (Ben-David et al., 2010) — a technique which adapts machine learning models trained on a domain with a sufficient amount of labelled data to a domain with little labelled data. The method is likely to be suitable for adapting a part-of-speech tagger model to primary care text, but even if an adapted model is successfully developed, it is also very likely that an adapted chunking model should be developed as well. The prospect of developing not one but two layers of annotation of the target data set, made this approach seem like high-cost alternative, which may have required more time and resources than were available for the writing of this thesis. Nevertheless, it is a viable approach and could be part of the future development of this work. The use of domain adaptation for chunking alone was also considered, but it was not clear if that would be a successful approach given the stark differences in syntax between edited text and the targeted primary care text. In addition, such an approach would require some labelled data and creating manual annotation compatible with the types of chunks produced automatically through constituency parsing seemed difficult and somewhat risky compared to starting from scratch.

1.4 MAIN CONTRIBUTIONS OF THE THESIS

The main contributions of the research reported in this thesis are an approach to developing a system for extracting information from primary care text, the Harvey Corpus of clinical text, two sets of guidelines used for annotating the corpus, an approach to maximising agreement between human annotators, a method for using a standard part-of-speech tagging model in clinical text chunking, and a novel method of concept recognition for text with terse expressions.

The challenges of primary care text, and the limited applicability of standard NLP tools to this kind of text make the implementation of a prototype concept recognition system an important first step towards deeper and more flexible analysis of primary care data. Such a system could be of great benefit to epidemiologists and other clinical researchers. Fast, automated analysis of this type could also support better exploratory analyses of large samples of data.

Even though access to the Harvey Corpus is still very limited, if the corpus could be released under a research license, it could be of significant importance for the development of clinical NLP research. In addition, from a corpus linguistics point of view, the set of chunking guidelines are an important document as it is the first of its kind for the English language (previous chunking annotation having been produced only for edited text, and as a by-product of parsing).

Finally, the new approach to concept recognition points the way towards tackling similar issues in the processing of other kinds of terse text, such as hospital progress notes.

1.5 THESIS SUMMARY

This thesis presents the Harvey Corpus of primary care text notes, named after the sixteenth century English physician William Harvey[5], and a medical concept extraction system tuned to the specific language of the domain. Chapter 2 gives an overview of theory and research literature relevant to the topics and aims of the thesis. It discusses other available corpora inside and outside the clinical domain, corpus annotation, relevant machine

5 The first accurate account of blood circulation is attributed to William Harvey, and presented in his book *De Motu Cordis*, also known as *On the Motion of the Heart and Blood*.

learning methods including classification, word embeddings, clustering, and evaluation, as well as fundamental NLP processes and how they have been used for analysing clinical text. Chapter 3 traces the design and building processes of the Harvey chunking annotation guidelines and corpus, and the training of medical specialists to annotate text. It describes in detail the aims, rationale, and the process behind building the two resources, as well as issues encountered along the way. Chapter 4 describes an extensive series of experiments developing the optimal combination of an existing publicly available part-of-speech model, and a feature set for a chunking model. The experiments also explore the role of word representation, a broad variety of context features, common classification algorithms, and problems of parameter optimisation. Chapter 5 introduces a further, clinical concept layer of annotation of the Harvey Corpus produced using another set of guidelines, and the final step of the system: recognising clinical concepts. The discussion focuses on a comparison between the usual method of concept recognition, and a novel approach that exploits the specific characteristics of the kind of text being processed. The chapter concludes with an evaluation of the entire system in real world conditions. Finally, Chapter 6 summarises the experimental results and achievements of this work, relates its contributions to various practical aspects of science and industry, discusses its limitations, and proposes directions for future work.

2

BACKGROUND & RELATED WORK

The medical concept recognition at the centre of this thesis is an information extraction (IE) task, focusing on recognising and classifying specific types of clinical concepts; as such it is similar to named entity recognition (another information extraction task) in the types of linguistic constructions they both target and partial overlap of some entities. For example, they both target dates and various types of measures, as well as different types of noun phrases. The most significant difference is perhaps that named entity recognition includes all kinds of named entities, i.e. names of places, objects and people, while medical concept recognition focuses on entities that are relevant for the medical narrative of a text, like drug and disease names, symptoms, procedures, and others. The two tasks are also approached in similar ways using virtually the same sequence tagging techniques. As these techniques rely on the output of a number of lower level NLP analyses and techniques, it is essential to review the relevant theory and literature for everything that was used or considered during the studies described in the following chapters, in order to determine their relevance to the primary care notes domain.

Regardless of the chosen methodology, adapting existing processing to a new domain requires a certain amount of labelled data. The data is needed for validation purposes if rule-based, unsupervised or cross-domain machine learning methods are used, and for both training and validation in the case of supervised machine learning methods. Section 2.1 introduces the notion of text corpora, some of the key issues in corpus development, and the most notable corpora in the biomedical and clinical domains. A particularly important aspect of corpus development and an objective measure of annotation quality is inter-annotator agreement (IAA). Section 2.2 surveys the different IAA metrics, explaining the choices made for the annotation processes in Chapter 3 and Chapter 5.

Machine learning methods dominate most parts of modern NLP, as well as the approaches taken in this thesis. Therefore, the chapter reviews the fundamental concepts and theory in order to facilitate the description of the methods discussed in the following chapters. Section 2.3 gives a shallow overview of classifiers, the broad feature engineering process, and recent research word representation clusters and embeddings. It also describes evaluation techniques commonly applied in classification tasks, and the domain adaptation technique in machine learning.

The basic principles and issues behind the various NLP processing steps leading up to information extraction are the culmination of this chapter. A typical IE system consists of a text segmentation step, optionally a word normalisation stage, a POS tagger, a chunker or a full syntactic parser, and various high level IE processes, such as named entity recognition and coreference resolution. Section 2.4 gives a description of each step and a brief account of relevant research literature, while Section 2.5 provides a more specific overview of NLP challenges, methods and research in the domain of clinical text.

2.1 CORPORA

The field of corpus linguistics is based on the assumption that linguistic theory can be abstracted or induced in an empirical way from large samples of recorded natural language usage, also known as text corpora. The authors of the Brown corpus (Kučera and Francis, 1967), one of the most notable early resources of its kind, stated that it should "offer useful material for the development and improvement of statistical procedures of linguistic analysis and will make possible the construction of more satisfactory mathematical models of language."

With advances in computing, larger language corpora have underpinned the development of modern natural language processing, especially so using statistical models. However, having a large enough corpus is not sufficient to solve all problems in NLP, as there needs to be a good match between the text in the corpus and the one to be processed. The reason for this dependency is that corpora are usually annotated with information that helps analyse a particular type of language, but more importantly it allows the induction of such information from unseen text based on some automatic processing. Thus the usability of corpora is still limited by various aspects of the language type they were sampled

from. Their characteristics remain dependent on the frequency distribution of word types, syntactic constructions, and other language patterns. However, there are some measurable qualities of corpora that may play a role in their applicability to certain tasks. McEnery and Hardie (2012) define the *representativeness* of a corpus as "one that is sampled in such a way that it contains all the types of text, in the correct proportions, that are needed to make the contents of the corpus an accurate reflection of the whole of the language or variety that it samples." They further define a corpus as *balanced* if "the relative sizes of each of its subsections have been chosen with the aim of adequately representing the range of language that exists in the population of texts being sampled." The two qualities are somewhat related, because a representative corpus has to be balanced, but a balanced corpus is not necessarily representative (Temnikova et al., 2014). For example, the British National Corpus (Clear, 1993) may have been representative and balanced at the time of its creation, but currently it is only balanced.

Despite the properties described above, comparing corpora in an objective and quantifiable manner still remains an open issue in the field of corpus linguistics (Kilgarriff, 2001). Therefore improving or adapting corpora to a new domain remains difficult and somewhat of a "black art". Adapting statistical language models to new domains, however, is still a necessity in NLP, because it remains the best means of developing and validating methods and models adapted to a new domain.

Beyond the language qualities of corpora, the types of annotation they are equipped with are another one of their important aspects. Even though corpora can be used to explore language characteristics on a large scale and derive or support linguistic theory through empirical methods, their main purpose in modern natural language processing is mainly as gold standard resources to be used for training statistical models and performance validation. Thus, corpora often have more than one annotation layer to serve different NLP tasks. The layers can be both manually created by human annotators, as in the cases of the Penn Treebank (Marcus et al., 1993) and the Prague Dependency Treebank (Bejček et al., 2013), or automatically generated by an annotation tool, as in the case of the British National Corpus (BNC). Treebanks are a common format of annotated corpora, which, as suggested by the name, contain (both constituency or dependency) parse tree annotations as well as the necessary parts of speech, and sometimes additional annotation.

Creating manually annotated corpora can be an arduous and convoluted process, depending on the annotation type and complexity. Each annotation process requires the selection or design of a set of annotation guidelines to set the course of the process. Some annotation formats like the Penn Treebank (Marcus et al., 1993) have become a de facto standard, and are frequently reused or modified by other corpora (Ritter et al., 2011; Derczynski et al., 2013), but very few are suitable for cross-language usage (Petrov et al., 2012). Strategies such as double-blind annotation are often used in corpus building in order to ensure the reliability of the data and to minimise the human bias of the annotators. The agreement between annotators is usually a good measure of the overall reliability of the annotation. It is usually reported using special coefficients, accounting for chance agreement, which are discussed in detail in Section 2.2.

Even though a great variety of corpora have already been created and there are some established good practices (Roger Garside and Geoffrey Leech and Anthony McEnery, 1997; Wynne, Martin, 2005), it is still difficult to identify firm rules about how to build or evaluate corpora in an objective and consistent way. The rest of this section briefly presents the more important corpora in the biomedical (Section 2.1.1) and the clinical (Section 2.1.2) domains, giving some details on how and why they were built and what data was used in the process.

2.1.1 *Biomedical Corpora*

The term *biomedical corpora* is generally used to refer to collections of text on topics in the life and biomedical sciences. Typically, they may contain a very wide range of studies and annotation types, but generally keep to sources of scientific writing commonly found through MEDLINE and PubMed. The rich terminology of such texts along with some semantic differences presents some difficulties for their processing with tools developed for general domain text, such as news articles. Given the similarities discussed above, it is important to review the corpora with more widely recognised impact in the field. In addition, Verspoor et al. (2012) provide a link to a nearly comprehensive list of publicly available biomedical corpora [1].

1 http://compbio.ucdenver.edu/ccp/corpora/obtaining.shtml

GENIA (Ohta et al., 2002) and GENETAG (Tanabe et al., 2005) are two of the best established and widely applied biomedical NLP resources. They both contain protein and gene annotation, providing a solid base for Information Extraction (IE) research. GENIA was manually annotated by domain experts using an ontology developed in parallel with the annotation process. Tanabe et al. (2005) used annotation guidelines that provided several related alternatives for each annotated protein or gene. This allowed a more structured partial matching for the evaluation of the entity recognition models it aimed to develop. They used a Naïve Bayes classifier to determine the likelihood of a document containing a gene or protein name. The selected documents were tagged using the AbGene tagger (Tanabe and Wilbur, 2002) and finally the annotations were manually transformed into GENETAG annotation by three domain experts. The Colorado Richly Annotated Full-Text (CRAFT) corpus described by Cohen et al. (2010) is a more recent resource that comprises 97 Open Access journal articles with syntactic, coreference, and concept annotations. Initially coreference was annotated using a modified version of the OntoNotes guidelines (Hovy et al., 2006). Later Verspoor et al. (2012) added syntactic annotation following the Penn Treebank annotation guidelines (Bies et al., 1995) and the BioIE addendum (Warner et al., 2004). At the same time, concept annotation was added to the corpus, identifying mentions of nearly all concepts from nine prominent biomedical ontologies and terminologies (Bada et al., 2012).

Another group of biomedical corpora worth mentioning were created through the BioCreative series of challenges. Most of the challenges in the series focus on protein-protein interaction extraction (Hirschman et al., 2005; Morgan et al., 2008; Krallinger et al., 2008; Leitner and Krallinger, 2010) as well as other types of relation extraction (Wei et al., 2016).

2.1.2 *Clinical Corpora*

Clinical text is written by medical practitioners in a clinical setting, describing interviews with patients, their medical history and pathology, medical findings established during interviews or procedures, and others. The spectrum of texts covered by the term “clinical” is, in fact, quite wide and can vary significantly in content, length, and style. For example, the GP notes central to this thesis are generally short with terse expressions, and difficult

to understand language, while internal research reports can resemble biomedical texts in both length and writing style.

During the past ten years a number of *clinical corpora* have been developed by the NLP community, thereby facilitating a great number of studies in the area (see Table D.3 for a representative list). Although the corpora have favoured longer text with more standard language such as admission and discharge summaries, other types of documents are also commonly included in the studies, e.g. progress notes and radiology reports.

Due to the difficulty in getting access to these kinds of data, shared tasks and challenges have played an important role in the development of the field, providing relatively easy access to the same resources to a wider range of scientists. Perhaps the most notable such enterprise is the i2b2 series of shared tasks and challenges, which also included a community annotation task. Uzuner et al. (2010b) present a set of guidelines for the annotation of a list of seven attributes associated with medications in discharge summaries. The guidelines were developed through an iterative process during which a group of students annotated a small number of discharge summaries and provided feedback used for the next revision of the guidelines. The guidelines were used in the i2b2 community annotation experiment, comparing the inter-annotator agreement (measured in F_1-score) of community annotator teams and expert annotator teams. The authors found that the IAA of the community teams was comparable to that of the experts, and concluded that involving the community in fairly complex annotation processes is an acceptable alternative to using expert annotators. The second part of the task was to automatically extract medication information (Uzuner et al., 2010a). The rest of the i2b2 challenge corpora were provided to the community in order to promote research in particular areas. Uzuner et al. (2007b) present an evaluation of the participating automatic de-identification systems, trained and evaluated on a corpus of 889 de-identified discharge summaries. A subset of that corpus containing 502 summaries was also annotated with patient smoker status for the purposes of one of the challenge subtasks (Uzuner et al., 2007a). Another i2b2 challenge was aimed at identifying obesity and its comorbidities in clinical text using a corpus of 1,237 discharge summaries (Uzuner, 2009). A subset of this corpus was later annotated with medical concepts and relations pertinent to congestive heart failure as part of the PhenoCHF corpus (Alnazzawi et al., 2014). The 2010 i2b2 challenge focused on identifying medical concepts, assertions, and relations (Uzuner et al., 2011). The organisers provided

the participants with 871 discharge summaries suitably annotated. Finally, a corpus of 310 discharge summaries annotated with temporal relations was provided for the latest i2b2 challenge (Sun et al., 2013a). The data annotation of all challenge corpora used two independent annotators and an adjudicator when possible. However, it is interesting to note that the adjudicators of the last challenge corpus were also allowed to edit or remove annotations in cases where the other annotators disagreed.

Other shared tasks have focused on document level annotation of clinical corpora. The Medical Records track of TREC 2011 and 2012 used 17,264 clinical documents of various types from the University of Pittsburgh NLP repository for a topic modelling task (Voorhees and Hersh, 2012). Pestian et al. (2007) present a small corpus of radiology reports annotated with ICD-9-CM codes.

Wang and Patrick (2009) present a small corpus of 311 admission summaries (45,953 tokens, 13,576 annotations), annotated with ten types of concepts based on SNOMED-CT. The guidelines were developed jointly by linguists and clinicians who annotated ten notes together. The guidelines were further refined in five iterations of annotation and analysis during each of which further five notes were annotated. The guidelines were completed once the inter-annotator agreement reached stable levels, at which point the real annotation began involving two computational linguists with some medical knowledge.

The CLEF corpus is another prominent clinical text resource (Roberts et al., 2008, 2009). It was developed to assist the development and evaluation of an IE system as part of a larger framework for the capture, integration and presentation of clinical information. The corpus includes 565,000 de-identified records of 20,234 deceased patients of the Royal Marsden Hospital oncology centre. An annotation scheme was developed using a cyclic process of annotating, analysing and improving. The records were first annotated by two medical domain experts and then the two sets of annotations were adjudicated by a third medical expert.

A few studies have focused on dealing with annotation supporting core NLP tasks such as part-of-speech (POS) tagging and syntactic parsing of clinical text. Pakhomov et al. (2004) describe the annotation of 271 clinical notes (100,650 tokens across 7,299 sentences) using the Penn Treebank guidelines (Santorini, 1990), achieving 87.95% average agreement of POS tagging annotation between three medically trained annotators calculated using Cohen's *kappa* (Cohen, 1960). More recently, Fan et al. (2011) presented two sets of 25

annotated progress notes from Kaiser Permanente Southern California (KPSC) and the University of Pittsburgh Medical Center (UPMC), a subset of the i2b2/VA challenge. They were annotated with POS tags for the purpose of developing and evaluating POS tagging models. The corpus comprises 31,400 tokens in 3,283 sentences annotated using a modified version of the original Penn Treebank part-of-speech tagging guidelines (Santorini, 1990). A following study on part of the same data presented a set of guidelines for syntactic parsing of ill-formed clinical sentences and a Treebank of 1,100 syntactically annotated sentences from the i2b2/VA challenge (Fan et al., 2013). The presented guidelines are an extended version of the Penn Treebank II bracketing guidelines (Bies et al., 1995). They were modified to help the annotators handle the non-canonical language of clinical text by flattening certain syntactic constructions, introducing a mechanism for handling omitted words, amongst other issues. The authors report IAA F_1-score reaching 93 on the final set of 450 sentences, and parsing accuracy reaching 81 using a statistical model trained on mixed data (newspaper and clinical text). Another syntactically annotated clinical resource is the MiPACQ corpus of Mayo Clinic pathology notes presented by Albright et al. (2013). The corpus consists of 127,606 tokens of text related to colon cancer annotated with POS tags and constituency parsed trees using a version of the Penn Treebank guidelines adapted to clinical text with some additional non-terminal nodes, e.g. for dropped subjects. In contrast with other corpora, the MiPACQ corpus was automatically annotated for the most part using existing tools (Coden et al., 2005; Bikel, 2002), and then corrected according to the guidelines, while only a small part was double blind annotated. Additionally, the MiPACQ corpus was annotated with semantic roles in the style of PropBank (Palmer et al., 2005), UMLS entities, and syntactic dependencies. The dependency annotation was generated through conversion from the constituency annotation using the Clear converter described by Choi and Palmer (2010) with some modifications to match the Stanford typed dependencies representation (De Marneffe and Manning, 2008). Xu et al. (2011) manually annotated 50 randomly selected sentences from the i2b2-2010 data with constituency parse trees using the original Penn Treebank bracketing guidelines with some additional examples from clinical text, which were developed through an iterative process of annotation, discussion, and guideline correction. Finally, the latest i2b2 challenge (Stubbs et al., 2015a,b) presented a corpus of 1,304 medical records for 296 diabetic patients where all protected health information (PHI) had been removed and replaced with

realistic surrogates. The data was distributed with PHI annotation for the de-identification track (Stubbs and Uzuner, 2015a), and heart disease risk factors annotation for the other track (Stubbs and Uzuner, 2015b).

2.2 INTER-ANNOTATOR AGREEMENT

All hand-annotated data resources, even ones made with exceptional skill come with the possibility of human error. That possibility decreases, but does not disappear with annotation of the same data by multiple annotators. It is virtually impossible to determine the existence of errors with absolute certainty, because all annotation that can serve as ground truth is also man made. However, the agreement between the two annotators can be measured using the same annotation guidelines on the same data. The assumption is that if different annotators agree in the categories they assign to the items in the data, they perform consistently, which is evidence of a similar understanding of the annotation guidelines, and ultimately of the validity of the annotation scheme, which is how well it captures the "truth" of the phenomenon being studied (Artstein and Poesio, 2008).

Inter-annotator agreement (IAA) can be measured in a variety of ways depending on the annotation setting and goals. Scott (1955) defines *percentage agreement* or *observed agreement* (A_o) as "the percentage of judgments on which the two analysts agree when coding the same data independently". However, observed agreement estimation does not account for chance, as also noted in the same article and illustrated by Artstein and Poesio (2008) with the following example. Consider the independent random classification of data items by two annotators using an annotation scheme. If the annotation scheme has two labels, then the annotators will agree on half of the items; if it has three labels, they will agree on one third of the items. Therefore, the observed agreement measurement is biased towards annotation schemes with fewer categories (labels), which makes the measure unsuitable for comparing different annotation schemes. Artstein and Poesio also note that observed agreement does not account for the distribution of items across categories, which greatly influences what can be perceived as high or sufficient level of agreement. Carletta (1996) gives the following example: if both annotators were to use one of two categories, but use one of the categories 95% of the time, we would expect them to agree 90.5% of the time (0.95^2 + 0.05^2 , or, in words, 95% of the time the first annotators chooses the

first category, with a 0.95 chance of the second annotator also choosing that category, and 5% of the time the first annotator chooses the second category, with a .05 chance of the second annotator also doing so). Given perfectly plausible cases like this, it is clear that there cannot be a single standard value for good or even acceptable *observed agreement* that can be used across different studies.

2.2.1 *Chance Corrected Agreement Coefficients*

The most popular inter-annotator agreement coefficients that correct for chance are based on the same idea of estimating the expected *agreement by chance* (A_e), and then comparing the observed agreement beyond chance to all the available agreement beyond chance. The former is calculated as the difference between the observed agreement (A_o) and the probability of agreement by chance, while the latter is the difference between 1 and A_e.

$$S = \pi = \kappa = \frac{A_o - A_e}{1 - A_e} \tag{2.1}$$

The coefficients S (Bennett et al., 1954), π (Scott, 1955), and κ (Cohen, 1960) use the formula in Equation 2.1, which reflects the above mentioned idea of accounting for agreement by chance.

$$A_e^S = A_e^\pi = A_e^\kappa = \sum_{k \in K} P(k|c_1)P(k|c_2) \tag{2.2}$$

The coefficients differ in the way A_e is estimated — more precisely, in the estimation of $P(k|a_i)$ in Equation 2.2, which represents the joint probability of annotators c_1 and c_2 independently assigning an item to an arbitrary category k from the set of categories K (Zwick, 1988; Hsu and Field, 2003; Artstein and Poesio, 2008).

BENNETT'S S coefficient assumes that a uniform distribution would be obtained if annotators were operating by chance alone. That implies that all annotation categories are equally likely, meaning $P(k_j|c_m) = P(k_l|c_n)$ for any two annotators c_m, c_n and any

two categories k_j, k_l. The expected chance agreement for the S coefficient is then defined as:

$$A_e^S = \frac{1}{|K|} \tag{2.3}$$

where $|K|$ is the number of items assigned to class k. This definition rewards the usage of a larger number of categories, because as $|K|$ grows A_e^S gets smaller, which means the chance agreement for S also gets smaller (Scott, 1955; Artstein and Poesio, 2008). This could be a problem when the distribution of categories over the annotated dataset is not uniform. Extremely fine-grained POS tagsets with many tags occurring very rarely or not at all in real life data, would be heavily favoured by the S coefficient compared to a simple tagset with under a dozen POS tags.

SCOTT'S PI coefficient tries to address the limitations of S's uniformity assumption through estimating the prior distribution of the annotation categories using the behaviour of the annotators. The idea was first proposed by Scott (1955) based on the assumption that there is an underlying distribution of the categories that governs the random assignment of items into them by the annotators. Under this assumption, the probability $\hat{P}(k)$ of an arbitrary item being assigned to a category k can be estimated using the observed probability of assignment to k, defined as the number of items assigned to k by both annotators normalised by the total number of assignments made by the two annotators. Scott does not account for any individual annotator bias, thus the probability of an item being annotated with a category k by any annotator is equal to $\hat{P}(k)$:

$$P(k|c_1) = P(k|c_2) = \hat{P}(k) = \frac{n_k^a}{2n} \tag{2.4}$$

where n_k^a is the number of items assigned to category k by both annotators and n is the total number of annotation items.

Assuming that each assignment of an item to a category is independent from other assignments, the probability of two annotators agreeing by chance can be estimated as

the joint probability of each of them randomly assigning that category summed over all categories.

$$A_e^{\pi} = \sum_{k \in K} \hat{P}(k) \cdot \hat{P}(k) = \frac{1}{4n^2} \sum_{k \in K} n_k^2 \tag{2.5}$$

Considering the chance agreement estimation in Equation 2.5, it is easy to show that $A_e^S \leqslant A_e^{\pi}$, as well as that $S \geqslant \pi$, with the equality being achieved only with a uniform distribution of annotation categories.

COHEN'S KAPPA coefficient assumes that each annotator has their own individual *bias* which is reflected in the prior distribution that governs the random assignment of items to categories (Cohen, 1960). Thus $P(k|c_i)$, the probability that the annotator c_i will put an arbitrary item into the category k, can be estimated as $\hat{P}(k|c_i)$, the observed proportion of items assigned to k by annotator c_i compared to the total number of items n:

$$P(k|c_i) = \hat{P}(k|c_i) = \frac{n_{c_i,k}}{n} \tag{2.6}$$

Making the same independence assumptions about chance agreement as for π, the chance agreement for κ can also be estimated by the joint probability of each annotator assigning an arbitrary item to category k.

$$A_e^{\kappa} = \sum_{k \in K} \hat{P}(k|c_1) \cdot \hat{P}(k|c_2) = \sum_{kinK} \frac{n_{c_1,k}}{n} \cdot \frac{n_{c_2,k}}{n} = \frac{1}{n^2} \sum_{k \in K} n_{c_1,k} \cdot n_{c_2,k} \tag{2.7}$$

Based on Equation 2.7 it can be shown that given the same set of annotated data, $A_e^{\pi} \geqslant A_e^{\kappa}$ as well as $\pi \leqslant \kappa$.

2.2.2 *Weighted Agreement Coefficients*

One of the greatest limitations of the chance corrected coefficients presented in the last section is their equal treatment of disagreement. That quality is suited for category sets with mutually exclusive members, but often NLP requires much more fine-grained approaches to annotation, which include categories with much in common, differing in just one particular aspect. For example, the Penn Treebank tagset (Santorini, 1990) focuses on the number aspect of nouns, and whether they are common or proper nouns, thus including four noun POS tags. If two annotators choose different noun tags, then that could be interpreted as mild disagreement, while if one annotator chooses a noun tag and the other an adverb tag, the disagreement could be interpreted as severe. In other cases such as annotation of coreference chains (used in coreference resolution), having different levels of agreement is absolutely necessary. A difference in one member between two sets, or even a difference in the number of members renders the sets different, thereby making the IAA coefficient extremely conservative and not as useful. The rest of this subsection discusses Krippendorff's α and Cohen's weighted κ coefficients, which aim to reflect different levels of disagreement.

KRIPPENDORF'S ALPHA coefficient was introduced by Krippendorff (1980, 2004) for the purposes of content analysis, but it has since been applied in a variety of cases where two or more methods of data generation are applied to the same set of items, and the reliability of the resulting data needs to be measured. It can be applied to data with any number of annotators, containing any number of categories, scale values, or measures (Krippendorff, 2011). It can also handle missing annotations, where not all items need to be annotated by the same number of annotators. It differs from other coefficients, which are all based on the formula in Equation 2.1, as in its general form its calculation is based on observed and expected by chance disagreement measures:

$$\alpha = 1 - \frac{D_o}{D_e} \tag{2.8}$$

Krippendorff (2011) gives detailed instructions on calculating α with different kinds of data in four steps. The first step is to create a reliability data matrix of all annotations organised by items and annotators (see X in *notation*). In step two the data in the reliability matrix is tabulated in a category pair coincidence matrix — a square matrix containing the coincidence value for each combination of annotation categories. The coincidence value of a category pair p-q for an annotation item i is defined as the number of times that category pair exists in all assignment pairs for i, normalised by m_i - 1, where m_i is the number of annotators that have assigned a category to item i. Thus the overall coincidence value for p-q is the sum of the item coincidence values over all annotation items:

$$Z_{p,q} = \sum_{i \in I} \sum_{p \in C} \sum_{q \in C} \frac{\omega(p,q,i)}{m_i - 1}, \tag{2.9}$$

where ω is defined as:

$$\omega(p,q) = \begin{cases} 1: & X_{i,p} = X_{i,q} \\ 0: & X_{i,p} \neq X_{i,q} \end{cases} \tag{2.10}$$

The third step is to determine a difference function δ to apply an appropriate weight to each coincidence value. If no weights are to be used, that is two categories either match or they don't, δ can be defined as:

$$\delta(p,q) = \begin{cases} 1: & p = q \\ 0: & p \neq q \end{cases} \tag{2.11}$$

When using that difference function and given that all annotators assigned all items to a category, it can be noted that D_o is the exact complement of multi-π's A_o. In fact, in that case π is almost equal to α with a difference factor of $(n|C|-1)/n|C|$ (Artstein and Poesio, 2008). It should also be noted that in order for α to work properly the difference functions need to project zeroes, i.e. agreement, along the main diagonal.

The difference function and the coincidence matrix are then used in the calculation of D_o and D_e. The observed disagreement is represented as the sum of coincidence values for each category pair weighed by the square of its difference function, and normalised by the number of annotation items.

$$D_o^\alpha = \frac{1}{n} \sum_{p \in C} \sum_{q \in C} Z_{p,q} \cdot \delta(p,q)^2 \tag{2.12}$$

The function $\xi(p)$ is defined as the sum of all coincidence values of all pairs including a category p, that is the sum of all values in a row or column in Z:

$$\xi(p) = \sum_{q \in C} Z_{p,q} \tag{2.13}$$

The disagreement by chance for a category pair p-q can be defined as the product of the sums of all coincidence values of each of the categories weighed by the square of the difference function, and normalised by the total number of annotation pairs η. Using ξ that definition can be expressed in the following way:

$$D_e^\alpha = \frac{1}{\eta} \sum_{p \in C} \sum_{q \in C} \xi(p) \cdot \xi(q) \cdot \delta(p,q)^2 \tag{2.14}$$

The last step in the calculation is replacing Equations 2.12 and 2.14 in Equation 2.8.

COHEN'S WEIGHTED KAPPA presented by Cohen (1968) is an alternative weighted IAA measuring coefficient. In some respects, such as using the same general framework as in Equation 2.8 and assigning disagreement weights to each category pair, it is similar to Krippendorff's α, but in others it is quite different.

$$\kappa^w = 1 - \frac{D_o}{D_e} \tag{2.15}$$

The two major differences are that κ^w is designed to handle only two annotators as opposed to an arbitrary number, and its weights are not restricted in any way, i.e. there is no predetermined minimum or maximum weight — although agreeing pairs must have zero weights.

The observed disagreement for an item i is calculated as the weight of the pair of annotation categories it was assigned to by the annotators as in the following equation:

$$disagr_i = \vartheta(X_{i,c_1}, X_{i,c_2}), \tag{2.16}$$

where ϑ returns the disagreement weight of two categories p and q:

$$\vartheta(p, q) = \Theta_{p,q} \tag{2.17}$$

where $\Theta_{p,q}$ is the disagreement weight of the category pair p and q (values between 0 and 1). Then the overall observed disagreement is the mean disagreement of all items normalised by the observed maximum ϑ_{max}.

$$D_o^{\kappa^w} = \frac{1}{n \cdot \vartheta_{max}} \sum_{i \in I} disagr_i = \frac{1}{n \cdot \vartheta_{max}} \cdot \sum_{i \in I} \vartheta(X_{i,c_1}, X_{i,c2}) \tag{2.18}$$

Given Equation 2.18 we can set the disagreement weights in such a way that $D_o^{\kappa^w}$ will be the exact complement of A_o^{κ} similarly to what was previously shown for α and multi-π. To achieve that for κ^w, all disagreements need to be set to an equal weight, that is $\vartheta(k_p, k_q) = 1$.

The overall expected disagreement by chance $D_e^{\kappa^w}$ is estimated practically using a simplified version of the same approach as α. Instead of it being based on coincidence values, the expected disagreement for an arbitrary item being assigned to a category k by an annotator c remains as originally defined for κ in Equation 2.6. Thus the probability of annotator c_1 assigning an item i to a category k_p while annotator c_2 assigns the same item to a category k_q is $\hat{P}(k_p|c_1) \cdot \hat{P}(k_q|c_2)$, the joint probability of each of the annotators making the assignments independently. The overall expected disagreement is the sum of

the aforementioned joint probability normalised by the number of items, and weighted by the respective category pair disagreement across all ordered category pairs, and then normalised by the maximal weight ϑ_{max}.

$$
\begin{aligned}
D_e^{\kappa^w} &= \frac{1}{\vartheta_{max}} \sum_{p \in K} \sum_{q \in K} \hat{P}(k_p|c_1) \cdot \hat{P}(k_q|c_2) \cdot \vartheta(k_p, k_q) \\
&= \frac{1}{n^2 \cdot \vartheta_{max}} \sum_{p \in K} \sum_{q \in K} n_{c_1,q} \cdot n_{c_2,p} \cdot \vartheta(k_p, k_q)
\end{aligned}
\tag{2.19}
$$

Again, the disagreement weights can be set to the same level, and it should achieve a $D_e^{\kappa^w}$ value that is the exact complement of D_e^k as defined in Section 2.2.1.

2.2.3 *Pairwise Agreement Coefficients*

The higher the number of annotators agreeing on an assignment, the more reliable it can be considered to be. Therefore, generalised versions of the π and κ coefficients have been used in studies using more than two annotators. Fleiss (1971) described a version of Scott's π coefficient suitable for more than one annotator[2], and Davies and Fleiss (1982) introduced a version for κ. The coefficients are calculated using the formula in Equation 2.1, but with different estimation of the A_o and A_e values. Following the pairwise coefficient naming convention used by Artstein and Poesio (2008), the generalised versions are referred to as *multi-*π and *multi-*κ. Fleiss proposed that when there are more than two annotators, the estimation of inter-annotator agreement for an item i being assigned to a category k should be based on the ratio between the agreeing pairs of judgments and the total number of judgement pairs η. Thus the observed agreement for an item i is the sum of this ratio over all categories.

$$
a_i = \frac{1}{\eta} \sum_{k \in K} \binom{Y_{i,k}}{2}
\tag{2.20}
$$

2 The original paper by Fleiss (1971) referred to the new coefficient as *kappa*, and in subsequent literature it was referred to as *Fleiss' kappa* or K when referring to the definition by Siegel and Castellan (1988). However, in essence both definitions are generalised versions of Scott's π because of its chance agreement estimation method (Artstein and Poesio, 2008).

where $Y_{i,k}$ is the number of annotators that assigned item i to category k. The observed agreement is then calculated as shown below:

$$A_o = \frac{1}{n}\sum_{i\in I} a_i = \frac{1}{n\eta}\sum_{i\in I}\sum_{k\in K}\binom{Y_{i,k}}{2} \tag{2.21}$$

where η is the number of judgement pairs $\binom{n}{2}$.

Fleiss makes the same uniform distribution assumption for the estimation of chance agreement in multi-π as assumed in Scott's estimation of π. Therefore chance agreement is estimated using $\hat{P}(k)$, that is the proportion of items assigned to category k by all annotators out of the total number of assignments by all annotators:

$$P(k|c) = \hat{P}(k) = \frac{1}{n\cdot|C|}\sum_{c\in C} n_{c,k} \tag{2.22}$$

where n_k is the number of items assigned to category k by all annotators. Based on Equation 2.22 and given the assumption that annotators act independently, the chance agreement for multi-π can be calculated as the joint probability of all annotators assigning an item to the same category summed over all categories. Equation 2.5 (Section 2.2.1) constitutes a special case of estimating chance agreement with exactly two annotators, but it can be generalised if 2 is replaced by the number of annotators $|C|$ as shown below:

$${}_{\text{multi}}A_e^{\pi} = \sum_{k\in K}(\hat{P}(k))^{|C|} = \frac{1}{(n\cdot|C|)^{|C|}}\sum_{k\in K}\left(\sum_{i\in I} Y_{i,k}\right)^{|C|} \tag{2.23}$$

As was shown in Section 2.2.1, the only differences between IAA coefficients are in the way they estimate the agreement by chance variable in Equation 2.1. Equation 2.7 shows the agreement by chance for only two annotators, which is based on the joint probability of two annotators independently assigning an arbitrary item to the same category. To adapt

this probability for more than two annotators, the joint probability needs to be adjusted to account for all annotators.

$$_{\text{multi}}A_e^{\kappa} = \sum_{k \in K} \prod_{c \in C} \hat{P}(k|c) = \frac{1}{n^{|C|}} \sum_{k \in K} \prod_{c \in C} n_{c,k} \tag{2.24}$$

2.2.4 *Agreement on a Large or Unknown Number of Items*

Historically, inter annotator agreement, or inter-coder reliability, coefficients originated in the psychology and content analysis fields, before being suggested to the NLP community by Carletta (1996). They have since become the standard for measuring the reliability of gold standard resources for certain NLP tasks, but not all. Hripcsak and Rothschild (2005) raised the issue of intractability of using chance-corrected coefficients for studies with an unknown or very large number of negative or irrelevant items. Even though Hripcsak and Rothschild raise the issue of calculating κ in the context of information retrieval, here the issue is observed from the point of view of annotating multi-word entities in natural language text, i.e. considering tasks such as syntactic chunking and named entity recognition.

Before delving into why chance related coefficients are not suitable for multi-word entities, we should discuss what kind of data they are suitable for and why. Inter-annotator agreement is built upon the following three finite sets: a set of annotators, a set of annotation categories, and a set of items. For instance, these sets can be identified when creating part-of-speech tagged data — corpora contain a finite number of tokens (items), they are annotated by two or more annotators, using a set of part-of-speech tags (annotation categories).

When considering multi-word entity annotation, tag sets similar to part-of-speech tag sets can be defined, but coming up with a set of items of tractable size to assign categories to is not as simple. In fact, it all hinges on what are items (entities) defined to be. In the case of POS tagging, they are defined as tokens, but entities consist of an arbitrary number of words (or tokens) in a particular place in the text. So in a nutshell the important difference between POS annotation and the annotation of some sort of multi-word entities

is that the latter should also determine the boundaries of the items in the text as well as their categories. Given this information the number of items in an entity annotation task can be defined in at least two ways.

The naïve way is to adopt the strategy used for automatic tagging of entities of arbitrary size and position in text. That strategy uses a token-based tagset with prefixes for the beginning, inside, and outside of a tag. Its assumption is that annotators should agree at the token level. However, measuring agreement this way could be very misleading. Token level agreement measures would be unfair if the variance of the average annotation token span is large. In this case the overall agreement would be skewed towards the agreement on annotation categories with greater average token span. For example, if annotator A marked two annotations, one with token span 4 and one with token span 1, and annotator B agreed only with the first one, then the observed agreement would be 80%. Furthermore, even if all categories are of the same fixed size the naïve approach will still be overly positive as entity annotation usually does not cover all tokens in the text, rather only a small number of sub-sequences.

The second option of defining the items in an entity annotation task is what can be called the brute force approach. We can assume that the text spans between entity annotations are in fact outside-entity annotations. Given this assumption, items can be defined as a sequence of one or more tokens at an arbitrary position in the text. So during annotation an annotator would pick an item (by defining its boundaries) and a category for it at the same time; thus non-annotated text would automatically become outside-entity. Given that perception of the annotations, the sum of the token spans of all annotations (including outside ones) will always amount to the total number of tokens. Therefore, we can consider the total number of items n_ι as the sum of a series of stars and bars problems[3], where the cardinality of the distinct tuples k ranges from 0 to the number of tokens in the text n_τ. Feller (1968) popularised the use of stars and bars as a graphical aid to solving combinatorial problems, such as counting the ways to put n indistinguishable balls into k distinguishable bins. The stars and bars approach is used in the proof of the following theorem:

3 Stars represent objects and bars represent the divisions between them. A sequence containing n stars and m bars represents a single possible grouping of n objects into m groups.

Theorem 1 *For any pair of positive integers* n *and* k*, the number of distinct* k*-tuples of positive integers whose sum is* n *is given by the binomial coefficient* $\binom{n-1}{k-1}$.

Using Theorem 1 we can derive the following formula to calculate the total number of items:

$$n_\iota = \sum_{k=0}^{n_\tau} \binom{n_\tau - 1}{k-1} \tag{2.25}$$

Given the item count calculation method in Equation 2.25, any text beyond a few sentences will have a vast number of items. For a corpus of one hundred tokens the number of items is approximately equal to 6.34e+29. Such a great number of total possible items makes any agreement calculation unusable as it will virtually always be approaching 1 even when measuring a corpus with millions of disagreeing annotations.

So far in this subsection it was shown that defining items in the case of entity annotation in (two different) ways that approximate their original definition (see the beginning of Section 2.2) compromises the fairness of the very agreement coefficients for the calculation of which an item definition was needed in the first place.

An alternative approach offered by Hripcsak and Rothschild (2005) addresses this issue by using F_1-score to represent annotation reliability, i.e. inter-annotator agreement, in cases where the number of negative examples (and therefore their total count) is very large or unknown.

THE f_1-SCORE also referred to as the *f-score*[4] or the *f-measure*, is the established performance metric for information retrieval tasks. It is in fact the harmonic mean of *precision* and *recall* (Equation 2.26), which reflect respectively the fraction of retrieved instances that are relevant, and the fraction of relevant instances that are retrieved. The two metrics are also referred to as *positive predictive value* and *specificity* in the medical literature. There is a well established trade-off between the two ratios — systems that

4 This thesis uses f-score to refer to F_1-score unless it is necessary to specify the exact value of β for comparison reasons.

favour precision suffer in recall and vice versa — and the F_1-score seeks to reward systems that balance them.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.26}$$

Consider the confusion matrix of a binary classifier system with the following cells: *true positives*, *false positives*, *false negatives*, and *true negatives*. Using the confusion matrix the following formulae calculating precision and recall can be derived:

$$precision = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \tag{2.27}$$

$$recall = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \tag{2.28}$$

Although in most cases it is better to have a balance between precision and recall, in some cases it might be required to favour one of them for external reasons. In such cases the general F_β-score formula is used with β being a positive real number (Equation 2.29). The standard precision biased value for β is 0.5 ($F_{0.5}$), while 2 is the standard value if a recall bias is desired (F_2).

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}} \tag{2.29}$$

		ANN1	
		Positive	Negative
ANN2	Positive	a	b
	Negative	c	d

Table 2.1: Confusion matrix of the agreement categories for a two-class annotation task with two annotators (Rogot and Goldberg, 1966). Each cell represents unit counts.

Hripcsak and Rothschild (2005) showed that if the annotations of either annotator are assumed to be the "gold standard" and the others the output of a "system", the F_1-score of the "system" approaches κ[5]. To illustrate this we need to consider the F_1-score representation using a confusion matrix for a retrieval task (see Table 2.1).

$$F_{\beta=1} = \frac{2 \cdot \frac{a}{a+b} \cdot \frac{a}{a+c}}{\frac{a}{a+b} + \frac{a}{a+c}} = \frac{2a}{2a+b+c} \tag{2.30}$$

Using the same notation Hripcsak and Rothschild express κ as in Equation 2.31.

$$\kappa = \frac{2(ad-bc)}{(a+c)(c+d)+(b+d)(a+b)} \tag{2.31}$$

For very large numbers, however, Equation 2.31 can be used to show that κ approaches F_1-score. Consider Equation 2.32 where d is very large and therefore all other variables are insignificant in the context of the ratio that κ represents. This allows us to remove any product that does not involve d and then remove d itself from the numerator and the denominator. This shows that the larger d the closer is κ to F_1-score.

$$\begin{aligned} \kappa &= \frac{2ad-2bc}{ac+ad+c^2+cd+ba+da+b^2+db} \\ &\approx \frac{2ad}{ad+cd+da+db} \approx \frac{2a}{2a+b+c} \end{aligned} \tag{2.32}$$

However, will this still hold true for a multi-label task? To answer this one needs an understanding of how sequence labelling tasks are evaluated using f-score.

2.2.5 *MUC-7 Scoring for NER Annotation*

While it is easy to see how the elements of F_1-score are extracted from an information retrieval task, it is slightly less obvious how F_1-score can be calculated for inter-annotator agreement. The MUC-7 (Chinchor, 1998) scoring system for evaluating named entity recog-

5 The choice of the annotator used as a "gold standard" is irrelevant as the different cases would only swap the values for precision and recall but keep the same F_1-score.

nition systems uses six categories instead of the values from the contingency table, while still calculating precision and recall (see Section 2.4.6 for a discussion of named entity recognition). The output of one of the annotators is treated as the gold standard, and then annotations are categorised into *correct*, *incorrect*, *spurious* (not present in the gold standard), *missing* (not present in the candidate annotation), *partial* (partial matches), and *non-committal* (null fills generated by the candidate annotation that were also null in the gold standard). Using these categories, precision and recall are calculated as follows[6]:

$$\text{precision} = \frac{\text{COR}}{\text{COR} + \text{INC} + \text{PAR} + \text{SPU}} \tag{2.33a}$$

$$\text{recall} = \frac{\text{COR}}{\text{COR} + \text{INC} + \text{PAR} + \text{MIS}} \tag{2.33b}$$

These representations of *precision* and *recall* illustrate how different errors (also disagreement) contribute to the ratios. Annotation boundaries are more important than the label, so mismatch on that level is handled regardless of the labels through the notions of *spurious* and *missing* annotations. The label errors (disagreement) are put in the *incorrect* category. If we need to express this with the notation from Table 2.1, the MUC-7 f-score expresses b and c as SPU + INC and MIS + INC (in any order), while a is the COR category. One might be tempted to interpret the *incorrect* category as part of a that was not correctly labelled, but the reason that is not true is that the multi-label annotation tasks add an additional dimension to what constitutes a retrieved document in a retrieval task. This view of the task increases the number of documents in the universe by a factor equal to the number of types in consideration, i.e. the annotator will need to agree or disagree that document x is of type y for each document and each type. Therefore, κ approaches F_1-score also in the case of multi-label annotation tasks.

2.2.6 *Micro- and Macroaveraging of f-score Results*

When an annotation process involves more than one annotation class (category) each with its own set of labels, the question of performance averaging arises. So far the discussion considered measuring reliability using f-score fitting either one class or a single set of

6 The MUC-7 evaluation did not award any partial credit, i.e. all partial matches were considered incorrect.

mutually exclusive non-overlapping classes. This section presents two complementary ways of averaging f-score.

THE MACROAVERAGING approach gives equal weight to each class regardless of the size of its population (Manning et al., 2008). It is calculated as the arithmetic mean of the f-scores of each class or group of classes:

$$f_{macro} = \frac{1}{|Q|} \sum_{i \in Q} f(\text{COR}_i, \text{INC}_i, \text{PAR}_i, \text{SPU}_i, \text{MIS}_i), \tag{2.34}$$

where f calculates the f-score using precision and recall formulae in the style of Equation 3.1 or 3.2, while Q is the set of classes or groups of classes to be averaged over.

THE MICROAVERAGING approach gives equal weight to each item of each class or set of classes (Manning et al., 2008). It calculates the f-score based on a pooled set of countable categories (such as COR, the true positives) — in other words the items of each countable category are counted together across all averaged annotation categories or sets of annotation categories and then the sums are used to calculate the average f-score:

$$f_{micro} = f(\sum_{i \in Q} \text{COR}_i, \sum_{i \in Q} \text{INC}_i, \sum_{i \in Q} \text{PAR}_i, \sum_{i \in Q} \text{SPU}_i, \sum_{i \in Q} \text{MIS}_i) \tag{2.35}$$

As mentioned above, the two ways of averaging differ in whether they view classes or instances as primary. Macroaveraging gives equal weight to each class, whereas microaveraging gives equal weight to each instance (decision). This makes microaveraging biased towards the performance of classes with higher frequencies, while macroaveraging results are susceptible to giving too much influence to outlier classes. Manning et al. (2008) summarises this as microaveraged results really being a measure of effectiveness on the large classes, while macroaveraged results provide a sense of effectiveness on small classes.

Generally, it is a good idea to use both averaging methods when reporting results, but given the flaws described above, it is better to provide the class specific metrics as well.

2.2.7 *Agreement Calculation in the Context of This Thesis*

Even though this section provides an extensive review of inter-annotator agreement coefficients, it is important to highlight only two that will be used in the context of this thesis. F_1-score is used for the multi-token annotation tasks in Chapter 3, while Cohen's κ is used for the labelling of symptoms, drugs and diseases among noun chunks in Chapter 5. The use of Krippendorff's α and weighted κ were considered for the latter task in order to explore the effect of different penalties for disagreement between certain classes, but no experiments were reported eventually.

2.3 MACHINE LEARNING IN NLP

Machine learning is the scientific discipline that studies and develops algorithms that allow computers to learn from data, and then be able to make predictions or decisions based on that learning rather than following pre-specified instructions entered by a human. Machine learning methods are most often used in computing tasks whose scale or complexity prevents the design of an effective rule-based solution, which is the case for many NLP tasks.

In machine learning, classification is the problem of identifying the category of an observation from a set of categories. It is usually tackled as a supervised learning task, which means that it needs a set of correctly identified observations from which the statistical model is built. Its unsupervised counterpart is called clustering — a method of grouping observations into subgroups called clusters based on a similarity or distance function.

This section discusses the application of machine learning for particular NLP tasks that are relevant for the experiments presented in the following chapters of this thesis. The discussion focuses on part-of-speech tagging, chunking (shallow parsing), concept recognition, and document classification.

Most natural language processing tasks can either be framed as a straight-forward classification task, or they can be broken down into a sequence of such tasks. For instance, part-of-speech tagging, assigning of the correct part-of-speech label to each token in a sentence, is a sequence of clearly defined classification problems. Other tasks need to be

approached differently in order to present them in a way that will make the classification optimal for supervised learning. For example, chunking can be seen as two separate classification problems – one determining the borders of a chunk, and one its type. Considered in such a way, the task seems rather complex and difficult, but it could also be simplified to a problem similar to part-of-speech tagging, where instead of phrase border and type classification, the problem is framed as token classification using a BIO notation.

The rest of this section discusses commonly used classification algorithms (Section 2.3.1), designing feature sets used by a classifier (Section 2.3.2), semantic word representation techniques (Section 2.3.3), as well as common classifier evaluation methods (Section 2.3.4).

2.3.1 *Common Classifiers*

A machine learning classifier is the implementation of a concrete classification algorithm that uses a *model* to determine the *class* of an *instance* based on a *feature vector* generated from its properties. In supervised machine learning an *instance* is an observation whose class needs to be determined — a token that needs a POS tag, a text document whose topic needs to be determined, etc. The possible classes are a finite set of discrete values or categories. As the machine learning model involves "learning" and storing different aspects of the training data, it is difficult to provide a simple but precise definition of its contents and building. However, we can characterise it as a machine learning device extracted from the training data and used to classify unseen observations.

Machine learning algorithms are based on mathematical and statistical operations, so in order to feed them observations in a non-numerical form, such as text snippets, these observations need to be represented in a suitable way. Feature vectors are used as the "classifier-friendly" representation of observations. They are ordered sequences of individual measurable properties, features, of an instance. A feature may be expressed with values of various types (binary, integer, real, categorical) depending on the classification algorithm. The length of the feature vector may be arbitrary as long as it is the same for all instances, but the decision on its length is not without consequence. A richer (larger) feature set is usually beneficial for the classifier, but it can also mean that the data contains more irrelevant information, which is a drawback. Finding the balance between feature set size and feature importance is usually an empirical issue.

There is a wide variety of methods, algorithms, and techniques in machine learning applicable to some aspect of NLP. The following sections give a brief account of the most relevant machine learning methods to this thesis.

2.3.1.1 *Single Prediction Classifiers*

Classifiers that make predictions of the class of a single instance are the most common type used in NLP. This is so, because some NLP tasks require classification of independent instances, while others can be easily reformulated in order to fit that paradigm. In a simple approach to document classification the feature vectors are generated based on *term frequencies*, which are essentially the frequencies of the words occurring in the document. Classification accuracy is usually improved if the term frequencies are normalised. Normalisation approaches include tf-idf (Spärck Jones, 1972) and smoothing, but unless there is a known relation between instances, it always remains true that a document class is independent from other documents, including their class and content. Therefore, target documents may be classified one by one in any particular order.

However, in part-of-speech tagging, and all other token-based classification tasks, some of the target instances depend on each other — mostly ones that are near each other, for example, in the same sentence. That dependency is implemented through the feature vectors, which are constructed from what are known as *context features*. As it is impossible to distinguish between words with multiple potential meanings or grammatical roles without the context they are used in, the features that play a decisive role in such classification decisions are drawn from the context of the token instance. However, for some locally-dependent tasks (e.g. POS tagging or syntax parsing) context features are limited to a small window around the target token, which should not reach outside the sentence. Based on how feature vectors are constructed, token-based processing assumes that tokens from the same sentence should be processed together. The processing order used by the majority of classifiers is the order of reading of the language, i.e. left ot right for English, but there are exceptions. Church (1988) takes a right-to-left approach, while Giménez and Màrquez (2004) describe a tool that is able to process tokens from each direction, as well as to combine their outputs.

NAÏVE BAYES is a family of classifiers (Stigler, 1983; Rish, 2001; Manning et al., 2008; Wikipedia, 2015a) based on a conditional probabilistic model derived from Bayes' law (Equations 2.36 & 2.37).

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}, \tag{2.36}$$

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}} \tag{2.37}$$

These classifiers are called "naïve" because of the often unrealistic assumption that all features are conditionally independent from each other given a class. What that means is that each feature contributes independently of all other features regardless of any correlations that exist between them. The derivation of the naïve Bayes' classifier is based on the conditional probability of a class y_k given a feature vector X:

$$P(y_k|X) = \frac{P(y_k)P(X|y_k)}{P(X)} \tag{2.38}$$

Since the denominator is a constant, which does not depend on the class y_k, the equation above can be re-written as the joint probability of the class and the feature vector:

$$P(y_k|X) \propto P(y_k, X) = P(Y_x, x_1, ..., x_n) \tag{2.39}$$

At this point the assumption of conditional independence of the features allows the use of the chain rule of probability, and rewrite the joint probability in the following way:

$$P(y_k|X) = P(y_k) \prod_{i=1}^{n} P(x_i|y_k) \tag{2.40}$$

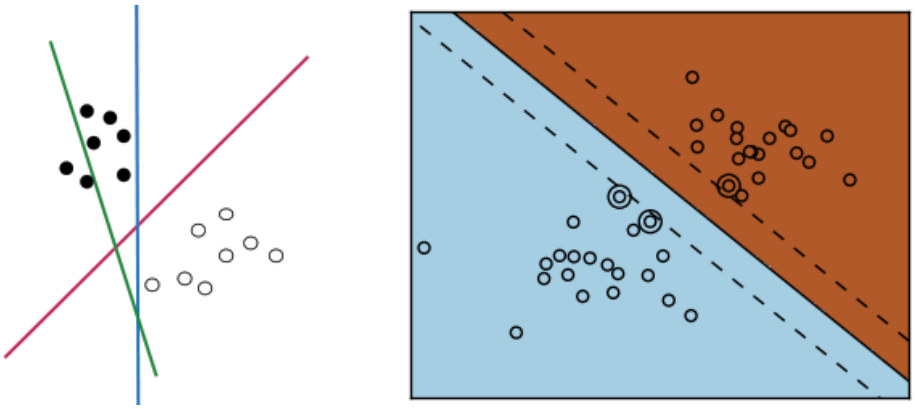
Using that conditional probability model, the classifier selects the class with the highest probability as its prediction $\hat{y}$, which is also known as the *maximum a posteriori* rule:

$$\hat{y} = \arg\max_{k \in K} P(y_k) \prod_{i=1}^{n} P(x_i|y_k) \tag{2.41}$$

There are three main versions of the naïve Bayes classifier, which differ from each other in the way they calculate the likelihood probability $P(x_i|y_k)$, or rather what distribution of the features they assume. A multinomial distribution is typically used for document classification tasks in which the feature vectors are essentially a histogram of term occurrences. A Bernoulli distribution (Kullback, 1935) is assumed, if the features are binary, which can be used in classifying short documents or tasks with context feature vectors. A Gaussian is usually not used in NLP as it is meant to handle continuous variables in the feature vector, which has few uses in language processing.

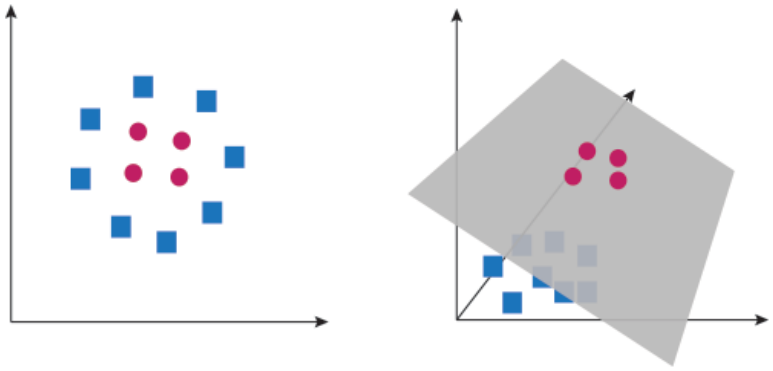
SUPPORT VECTOR MACHINES (SVMS) are machine learning devices that construct hyperplanes in a high-dimensional space, which can be used for binary classification. The intuition is that feature vectors are in fact data points in a p-dimensional space, and there exist $(p-1)$-dimensional hyperplanes that separate one class from the other (see Figure 2.1a). The margin is the distance between such a hyperplane and the nearest data points on each side, called support vectors. Linear SVM classifiers find the hyperplane separating a class from the rest with the maximum margin (See Figure 2.1b).

In some cases the maximum margin between classes is very small because of outliers that are very close to each other. Cortes and Vapnik (1995) introduce the soft margin variation of SVM classifiers, which optimises a trade-off between margin width and error penalty using a slack function. In other cases there is no linear classification solution, so the dimensionality of the feature vectors needs to be increased in a way suitable for classification. However, as feature vectors are already of high dimensionality, that may be challenging from a computational point of view. Boser et al. (1992) described a method that applied the "kernel trick" proposed by Aizerman et al. (1964) to maximum margin hyperplanes. Through that trick, non-linearly separable data can become linearly separable, thus easier for classification, in a higher dimensional space (see Figure 2.2).



(a) (b)

Figure 2.1: (a) Different possible class separations illustrated as hyperplanes dividing the data points. H1 does not separate the two classes, H2 does so by a very small margin, while H3 achieves separation with the maximum margin. Image based on Wikipedia (2015b); (b) Separation of two classes of data points by a hyperplane (unbroken line). Double circled data points indicate support vectors. Image by Haenel et al. (2013)



(a) Input space (b) Feature space

Figure 2.2: Illustration of the kernel trick. Images based on Wagner (2012)

Even though support vector machines are a very powerful tool, they have the limitation of being binary classifiers, and two classes is not enough for many tasks, especially in NLP. The solution to this limitation is problem binarisation and training of multiple classifiers. For example, part-of-speech tagging may use SVM classifiers by breaking down the multi-class classification task into either one-vs-rest decisions for each class, or one-vs-one decisions for each pair of classes. The rating is then done according to the SVM output function when using one-vs-rest classifiers, or according to the most wins when using one-vs-one classifiers.

MAXIMUM ENTROPY (MAXENT) classifiers (Malouf, 2002), also called *multinomial logistic regression* (Engel, 1988), generalise logistic regression classification to a problem

with more than two possible discrete outcomes. Logistic regression relies on a linear combination of observed features, usually continuous variables, and problem-specific parameters called weights to make a binary classification decision using the logistic function

$$F(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{2.42}$$

where β_0 is the bias, β_1 is a vector of weights, and x is the observed feature vector. Logistic regression does not assume independence of the observed feature variables, as Naïve Bayes does, and unlike other binary linear classifiers its output can be interpreted as a probability.

Multinomial logistic regression is represented as a set of independent binary regressions, which enforces an additional assumption of independence of irrelevant alternatives. This assumption states that the odds of a class being selected over another are independent of the existence of "irrelevant" alternative classes. For example, the relative odds of choosing a pet cat over a pet dog should remain the same if the possibility of a pet pony is introduced.

MaxEnt classifiers are often used in NLP as an alternative to Naïve Bayes since they do not make the naïve independence assumption, and there are a number of tasks to which they have also been applied successfully, for instance POS tagging (Toutanova and Manning, 2000; Toutanova et al., 2003) and chunking (Koeling, 2000).

2.3.1.2 *Structured Prediction Classifiers*

So far the discussion focused on methods of classification that deal with one decision at a time. This paradigm is suitable for some NLP tasks like document classification where documents are classified completely independently from each other, but it requires certain compromises in other cases such as POS tagging and parsing where classifications within the same sentence often depend on each other. There are ways to mitigate the effects of the assumption of independence in such cases by introducing dynamic features (using POS tags of already tagged tokens as context features), but they do not resolve the problem completely.

Structured prediction classifiers are a group of machine learning techniques that are able to predict structured objects, rather than single values. Many of them are based on

probabilistic graphical models such as Bayesian networks, but some are generalisations of other algorithms such as structured SVMs. A number of such algorithms have been used in NLP — hidden Markov models (Jurafsky and Martin, 2009, Chapter 6), maximum entropy models (Ng and Jordan, 2002), and conditional random fields (Lafferty et al., 2001) — the latter being one of the most widely used machine learning methods for sequential tagging tasks in recent years.

CONDITIONAL RANDOM FIELDS is a discriminative undirected graphical model whose nodes are made up of the disjoint sets of observed variables X (e.g. tokens), and label variables Y (e.g. POS tags). A structured prediction about a sequence is then modelled as the conditional distribution $P(Y|X)$. Lafferty et al. (2001) provide the following definition of conditional random fields (CRF):

> Let $G = (V, E)$ be a graph such that $Y = (Y_v)_{v \in V}$, so that Y is indexed by the vertices of G. Then (X, Y) is a conditional random field in case, when conditioned on X, the random variables Y_v obey the Markov property with respect to the graph: $P(Y_v|X, Y_w, w \neq v) = P(Y_v|X, Y_w, w \sim v)$ where $w \sim v$ means that w and v are neighbours in G.

Given this definition, the problem of inference in the case of general graphs is intractable, but in the special case of trees (and chains), inference is possible using algorithms similar to the forward-backward and Viterbi algorithms (Lafferty et al., 2001). The main advantage of CRFs over other probabilistic graphical models used for structured prediction is that they do not make strong independence assumptions (Lafferty et al., 2001). This allows the model to be optimised considering the whole observation sequence. Additionally CRFs avoid the bias of directed graphical models towards states with few successor states, also known as the Label Bias problem (Bottou, 2001; Lafferty et al., 2001).

Linear Chain CRFs are a widely used classification method in NLP which have been applied to POS tagging, chunking, and various entity recognition tasks (NER, gene recognition, medical concept recognition, etc.).

2.3.2 *Feature Engineering*

Machine learning classifiers require relevant aspects of language to be encoded in the format of a feature vector. However, deciding what those features should be is a problem in its own right. *Feature engineering* is the process of designing a suitable feature set for a particular task.

Token labelling tasks (e.g. POS tagging, chunking, and entity recognition) typically rely on features extracted from a context window around the token being classified. Common features are the surrounding tokens, and any available annotation — higher levels of analysis use lower levels, for example POS tags are used in chunking, chunks in named entity recognition, etc. Languages with rich morphology make use of morphological features, which could be of great importance, although they are of less consequence for English. There is also a wide variety of binary features like capitalisation, which could aid classification in particular cases.

Document classification on the other hand aims at using the most relevant subset of all tokens in the document regardless of their position, but using other methods such as frequency to weight the relative impact of features. Additionally, token features can be integrated with various types of linguistic analysis depending on the goals of the particular classification task. This may increase the number of features tremendously, and introduce noise to the classification process. *Feature selection* is the process of selecting the most relevant subset of features based on a metric that indicates their contribution to classification decisions (Guyon and Elisseeff, 2003).

In some cases the data from which features are extracted provides more information or has more variability than is needed for the purposes of a particular classification task. In such cases the features can be transformed into less variable form — a process called *feature canonicalisation*. For example, converting all words to lower case is a common practice in document classification as disregarding casing reduces the number of features and consolidates the word frequencies, thereby reducing training time and potentially improving performance. However, feature canonicalisation is applied on a case by case basis since for some tasks the discarded information may be of importance, e.g. word capitalisation information is crucial for named entity recognition.

2.3.3 *Word Representation*

Determining the meaning of words, phrases, sentences, and language as a whole is a well known problem in philosophy dating back to the works of Parmenides and Socrates. It also receives a fair amount of attention in 20$^{\text{th}}$ century philosophy through the work of some of its most influential philosophers Saussure (Culler, 1976) and Wittgenstein (2010). In computational linguistics there are two general approaches to the problem, one, referred to as *computational semantics*, seeks to compute meaning through a formalism based on formal logic (e.g. *Universal Grammar* by Montague (1974)), and the other, referred to as *distributional semantics*, represents meaning following Wittgenstein's dictum that meaning is use. This section briefly discusses distributional semantics, and two types of machine learning features derived through it.

2.3.3.1 *Word Embeddings*

One of the simplest yet accurate descriptions of the notion of distributional similarity can be expressed through the famous quote by John Rupert Firth stating that "[y]ou shall know a word by the company it keeps" (Firth, 1957). Lazaridou et al. (2014) illustrate this notion using the made up word *wampimuk* in the following example sentence:

`We saw a cute little wampimuk sleeping in the tree`

Given how it was used one can assume that a *wampimuk* is a living being with certain sympathetic features which is also able to climb a tree. Given sufficient additional examples, one can even come close enough to a full characterisation of the word's meaning. Thus the meaning of a word can be denoted by the set of contexts it occurs in. If all possible contexts are ordered, the meaning of a word can be represented by a vector, which effectively allows the learning of word representations from unlabelled data[7]. Such word representation vectors are sometimes referred to as *word embeddings*.

There are two broad classes of algorithms for building distributional representations for a single word. The first, commonly referred to as "counting" algorithms, was proposed by Grefenstette (1994). Such algorithms produce a model by counting the occurrences of a feature in the context of a word occurrence. Features are typically defined

7 As pointed out by Erk (2012) some authors choose to represent words as higher-order tensors, trees, or forests in order to capture more complex properties of their behaviour.

as other words or short phrases. The context of a word is considered to be the words occurring within a distance of k words in the same sentence. For example, in the sentence `Mary likes white dogs` the features of the word token `likes` would be the set {MARY, WHITE} if using a symmetric window of size one, or {DOGS, MARY, WHITE} if using a window of size two and single words as features. Alternatively, pairs of adjacent words can be used as features, in which case the features would be the set {WHITE_DOGS}.

The feature counts are typically weighed by a factor that reflects their informativeness, which is motivated by the intuition that not all contexts are equally important. For example, common words such as *the* and *to* can easily end up as contexts of most word entries given a wide enough window. Common methods for re-weighting the context counts include (positive) mutual information, log-likelihood ratio, and χ^2 (Evert, 2005).

The context vectors are typically of a very high dimensionality, which makes them difficult to work with in practice. Dimensionality reduction techniques, such as Singular Value Decomposition and Non-negative Tensor Factorisation, are commonly employed to address the issue (Turney and Pantel, 2010; de Cruys, 2010).

The second type of algorithms for building distributional word representations is driven by the same intuition as the "counting" approach — count, re-weight, reduce — however, instead of three consecutive processes these algorithms encode the desirable properties of the produced word vectors as a loss function, which is optimised using a neural network. There are two popular instances of that class of algorithms — WORD2VEC (Mikolov et al., 2013) and GLOVE (Pennington et al., 2014) — the work in this thesis draws on the former.

The size of the resulting embeddings produced by both types of algorithms typically ranges between twenty and several thousand. The common approach to using them for building machine learning models is to simply extend whatever feature vectors were built using regular feature extraction. In some cases, though, they have been used to completely replace regular features (Lebret et al., 2013).

2.3.3.2 *Clustering*

Clustering or cluster analysis (Driver and Kroeber, 1932) is the grouping of a set of objects in such a way that members of each group are more similar to each other than to members of other groups. The technique is most often used for exploring and analysing data in a visually accessible way. As clusters are "in the eye of the beholder" (Estivill-Castro, 2002)

there is no single definition of what a cluster is, or how one should be derived. There are a number of very different algorithms that produce clusters with a variety of qualities and suitable for different applications.

Word representation clusters are essentially thesauri based on distributional similarity. The words are grouped together based on a similarity measure computed from their context distributions. Here are some examples of such clusters produced by one of the most popular algorithms currently used in NLP (Brown et al., 1992):

```
{Friday Monday Thursday Wednesday Tuesday Saturday Sunday weekends
Sundays Saturdays}
{June March July April January December October November September
August}
{people guys folks fellows CEOs chaps doubters commies
unfortunates blokes}
{down backwards ashore sideways southward northward overboard aloft
downwards adrift}
{American Indian European Japanese German African Catholic Israeli
Italian Arab}
{liberal conservative parliamentary royal progressive Tory
provisional separatist federalist PQ}
{head body hands eyes voice arm seat eye hair mouth}
```

The algorithm is commonly referred to as Brown or IBM clustering. It is a hierarchical clustering algorithm based on an n-gram language model. Hierarchical means that the algorithm builds multiple layers of embedded clusters with different granularity. Although it has been previously applied to POS tagging (Ushioda, 1996), it only gained popularity in the NLP community once the computational costs of processing sufficiently large amounts of text became tractable, also enabling work on other word representation techniques (Turian et al., 2010).

There are a number of widely used word representation cluster resources, which were derived from huge corpora such as the RCV1 (Lewis et al., 2004) or the English Wikipedia, which have both very high numbers of words as well as high variation of their usage.

However, the word usage patterns in more specialised types of language, such as those in biomedical and clinical texts, are often not reflected in the word representations drawn from general text. Stenetorp et al. (2012b) present a study examining the effects of clusters and word embeddings on NER of biomedical text. They show that Brown clusters based on PubMed abstracts are more successful than those based on RCV1.

One of the most widely used set of clusters in POS tagging and chunking, the one used by the Stanford NLP tools, is based on another distributional similarity algorithm described by Clark (2003). Clark's clustering algorithm is commonly known as *Ney-Essen* clustering because it is an extension of the algorithm suggested by Ney et al. (1994). However, Clark's algorithm also integrates morphological information about the words. One of its main objectives was to be able to cluster less frequent words, which are generally more difficult to deal with when using purely data-driven techniques.

2.3.4 *Evaluation of Machine Learning Classifiers*

Estimating the accuracy of a model is important not only to demonstrate how it may perform on real world data, but also to provide a success measure that can be used to select the best configuration of parameters and features (Kohavi, 1995). Ideally the accuracy estimation method should have both low bias and low variance, but minimising both types of errors is subject to the bias-variance trade-off (Manning et al., 2008). Bias is the algorithm error caused by erroneous assumptions, which often happens if there are not enough features or training data. Variance on the other hand is caused by sensitivity to small fluctuations in the training set. This happens when the algorithm has "memorised" the data too well.

To make this trade-off, models need to be evaluated on unseen data. Since all datasets are finite, the simplest way to achieve this is to hold out some of the data from the training process and use it for evaluation. A potential flaw of this approach is that one or more of the classes of interest may be over- or under-represented, making the evaluation results less representative of the whole dataset. A very large dataset is less likely to have this problem, but since annotated resources are expensive to create, they are often relatively small. Thus validation techniques are used for assessing how the results of statistical analysis will generalise to an independent (real life) dataset. The evaluation methods should provide

the optimal partitioning of a small pool of data into independent subsets, which should be as large and as representative as possible.

Kohavi (1995) recommends k-fold stratified cross-validation as the best method for model development compared to repeated-learning testing (Zhang, 1993; Arlot and Celisse, 2010), also known as Monte-Carlo cross-validation, and bootstrapping (Efron, 1979; Efron and Tibshirani, 1997). In k-fold cross-validation (CV) the entire dataset is split into k equal subsets, each of which is in turn used as the validation set (while the rest are used for training). Stratified cross-validation ensures approximately the same proportion of instances of each class in each subset. In contrast, Monte-Carlo cross-validation repeatedly splits the data, selecting a predefined number of data points for the training set at random without replacement (meaning any data point can be selected only once per sampling), while the remaining are used for the test set. Even though there is no universally accepted proportion between training and testing data items (Arlot and Celisse, 2010), 10-fold CV (90:10) is the most widely used evaluation variety in NLP.

Bootstrapping, especially with high numbers of data items, is a computationally-heavy evaluation technique. The method divides the data pool randomly in two — a training and a test set — and then repeatedly resamples the two datasets with replacement. The resampling-evaluation process is repeated ideally as many times as the number of data items, but in cases where that is intractable, a lower number is selected instead.

The process of feature engineering typically involves following some theory or body of experience about what features may be useful, or simply searching for a combination through trial and error. In either case one needs to ensure that the best feature set is objectively evaluated on unseen data, which is likely to be overly optimistic if done using a simple training-testing split as discussed so far. Instead, a third subset has to be introduced for reporting evaluation results after a model has been selected. For example, in the case of k-fold CV, leaving an extra validation set is called inner cross-validation (Azzalini and Scarpa, 2012). The method divides the data pool into k+1 subsets and uses k of them to select a model via k-fold cross-validation, while the last one is used as a final validation set. Reporting the results of the final evaluation, rather than what was achieved during the model development, ensures the objective estimation of the model's performance.

2.3.5 *Domain Adaptation of Supervised Machine Learning*

Applying machine learning to data that does not match the initial training set often produces poor results. Domain adaptation of supervised machine learning is the task of adapting a statistical model to maximise its performance in a target domain where little or no labelled data is available, while there is another (source) domain with a much larger amount of training data available (Ben-David et al., 2010). There are a number of different ways of achieving that goal through different types of feature manipulations (Daumé III, 2007; Finkel and Manning, 2009; Ben-David et al., 2010; Schnabel and Schütze, 2014), but most methods do require some minimal amount of training data in the target domain.

2.4 BASIC NATURAL LANGUAGE PROCESSING

Natural language processing (NLP) is a fast moving academic field that involves both basic and applied research, with strong relationships to the fields of Computer Science (CS), Artificial Intelligence (AI), and Linguistics. As such, the tasks it involves could be divided into the two non-exhaustive groups: natural language analysis and natural language generation. This thesis is exclusively concerned with the former, and thus all references to NLP below actually refer to natural language analysis.

Many NLP applications comprise a sequence of lower-level basic tasks. This section briefly presents the basic NLP tasks, as well as information extraction which is especially relevant for the rest of the thesis. Note that some tasks, such as morphological analysis are omitted here as they are not particularly important for processing English language, but they do play an important role in other languages.

2.4.1 *Segmentation*

The most basic task in NLP is the breaking-up of text into segments that reflect the linear, sequential structure of language. Typically this involves two processes — *tokenisation* and *sentence splitting* (also known as sentence boundary detection). Tokenisation is the process of segmenting a sentence into tokens, which are mostly words and numbers, but

also punctuation and various other signs used in text. The two processes are sometimes carried out together, although more often sentence splitting is performed first.

Although there are implementations of tokenisation and sentence splitting using machine learning, simple generic rule-based approaches are the predominant solution for edited text, because of its regularity (Dridan and Oepen, 2012). Rule-based approaches are also preferable in specific cases like tweets where segmentation is particularly important (Gimpel et al., 2011).

2.4.2 *Spelling Correction*

Spelling correction is the task of recognising and correcting spelling mistakes. Kukich (1992) defines three types of error correction:

- **non-word error correction:** correcting errors, which result in non-words (e.g. *tere* instead of *there*)
- **isolated-word error correction:** correcting errors resulting in non-words, but only looking at the word in isolation
- **context-dependent error detection and correction:** detecting and correcting with the help of context spelling errors resulting in real words, e.g. *there* instead of *their* or *three*

These correction problems are generally tackled using four groups of methods: edit distance metrics (e.g. Levenshtein distance Levenshtein 1966, or the SOUNDEX algorithm O'Dell and Russell 1922), the noisy channel model (Shannon, 1948), machine learning classifiers, or machine translation.

The methods using edit distance metrics use dictionaries and a metric to rank the possible corrections of a word. The metrics are based on orthographic operations that convert a wrong string representation to a possible correction, e.g. character insertion, replacement, deletion, and transposition. On one the hand they have an advantage in being able to correct errors in isolated words, but on the other hand by themselves they do not consider the word context, thus cannot be used for errors with real word observations. Some notable studies using this method were presented by Pollock and Zamora (1984), and Wong et al. (2006).

The noisy channel model is a general approach to the problem of normalising noisy signal that has been applied in many fields and is one of the preferred techniques for text normalisation (Mays et al., 1991; Church and Gale, 1991; Brill and Moore, 2000; Toutanova and Moore, 2002; Choudhury et al., 2007; Cook and Stevenson, 2009). It assumes a scenario where a signal is sent through a noisy channel which alters it so that the received signal is different from the sent signal. The model seeks to identify the best version of the original signal given the received (observed) signal.

If text in need of normalisation is assumed to be the same text that was intended by the author, only written in another language, then it seems logical to emulate statistical machine translation for the purpose of text normalisation (Aw et al., 2006; Kobus et al., 2008). The strongest point of such approaches should be their ability to replace m words with n words from both sides during normalisation. Their drawbacks are that they are not robust, and they require aligned training data.

Finally, it is difficult to imagine word normalisation, as a whole, as a classification problem, although parts of it could be broken down to small tasks which can be handled by classifiers with high accuracy. Sproat et al. (2001) classify the types of normalisation that can be applied to tokens, while Lita et al. (2003) approach case normalisation as a classification task and use HMMs to tackle it. Han et al. (2012) use an SVM classifier to find out if correction is needed, and if so to select the best correction candidate based on a closed set of possible corrections generated using letter- and phoneme-based edit distance features.

2.4.3 *Part-of-Speech Tagging*

Part-of-speech (POS) *tagging* is the assignment of a label out of a predetermined set of POS tags to each of the tokens in segmented text. The difficulty of the task comes from the fact that often words have many possible parts of speech depending on the context they are used in. Therefore determining the most likely part of speech must take into account the surrounding words and their parts of speech.

A widely used tagset, The Penn Treebank tagset (Santorini, 1990), has somewhat over thirty tags (exact number depends on the version), although there are only ten basic parts of speech commonly used in English: *noun*, *verb*, *adjective*, *adverb*, *pronoun*, *preposition*,

conjunction, *interjection*, *numeral*, *article*. The additional labels come from incorporating different aspects of grammar like verb tenses, distinguishing between punctuation symbols, and accounting specially for certain words with idiosyncratic grammatical behaviour, such as the existential *there*. Generally, richer tagsets offer better solutions, but a trade-off exists between the tagset size and the difficulty of the tagging task. This makes using extremely fine-grained tagsets somewhat impractical from an automatic tagging point of view, while using oversimplified ones has limitations. There are cases such as processing tweets (Gimpel et al., 2011) or cross-language processing (Petrov et al., 2012), in which a limited tagset could be beneficial. There are also cases where a fine-grained tagset can yield better results, as is the case of the BulTreebank tagset for Bulgarian (Simov et al., 2004), which incorporates morphosyntactic information.

Although rule-based approaches have been used in the past, modern approaches to part-of-speech tagging put an emphasis on machine learning aided in certain cases by rules. The reason is that the problem setting of the task, given some assumptions, is a perfect classification task. Assuming that each POS tag is a class label in a classification process, and that the label of each word does not depend on the POS tag of its neighbours, classifiers such as Naïve Bayes (described later in this chapter) and Support Vector Machines (Vapnik, 1998) can be easily adapted to the task by using them to label tokens in order from left to right (or vice versa). Alternatively, structured classifiers such as CRFs do not need the independence assumption as they can determine the POS tags of all tokens in a sentence at the same time, optimising for the overall solution.

2.4.4 *Parsing & Chunking*

The syntactic analysis of a string of symbols according to a formal grammar is called *parsing*. It is a procedure used not only for natural language, but also for systems in which symbols obey structural constraints, such as computer programming and gene sequences. There are two common kinds of formal grammars used for natural language: constituency grammars and dependency grammars.

2.4.4.1 *Constituency Parsing*

A constituency parse represents the syntactic structure of a sentence as a tree consisting of a hierarchy of phrases (or constituents), with words at the lowest level. The types of structures that may be produced are defined by a grammar. One of the simplest kinds of phrase structure grammar is context-free grammar (Chomsky, 1957) shown in Example 2.1. Other notable constituency grammar formalisms are head-driven structure grammar, HPSG (Pollard and Sag, 1994), lexical function grammar, LFG (Bresnan, 2001), and tree-adjoining grammar, TAG (Joshi and Schabes, 1997). The tree that is produced by a constituency parser divides a text into phrases. The non-terminal nodes are types of phrases, while the terminal nodes are the words in the sentence. The edges have no labels and represent the hierarchical structure.

$$S \rightarrow NP\ VP \qquad NP \rightarrow NNP$$
$$VP \rightarrow VBZ\ ADJP \qquad ADJP \rightarrow JJ$$

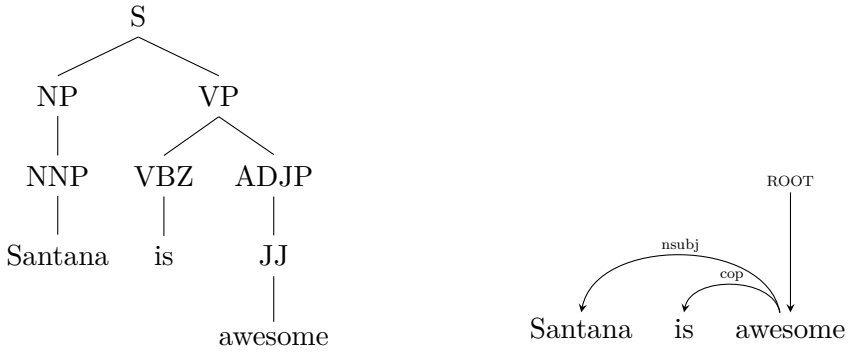
Example 2.1: Phrasal structure grammar used for the parse tree in Figure 2.3a.

Natural language is very complex, thus often allows more than one valid syntactic interpretation of a sentence. It is also immensely diverse, which makes the manual design of a complete grammar very difficult. Therefore, grammars are commonly constructed by inferring them from a syntactically annotated corpus called a *syntactic treebank* (e.g. the Penn Treebank, Marcus et al. 1993).

Probabilistic context-free grammars, PCFG, (Sarkar, 2011) can be inferred from treebanks, together with estimates for probabilities for the different rules of the grammar, in order to select the most probable parse tree out of all possible trees. While this approach calculates the probability of a parse tree as the joint probability of the rules that were used to construct it, the *history-based* approach (Black et al., 1992) is a generative method that takes the tree building process into account. It calculates the probability of a parse tree as the product of conditional probabilities of each building step given its *history*, i.e. the partial tree. Some notable history-based parser implementations are presented by Charniak (2000), Collins (2003), and Klein and Manning (2003).

One of the problems with PCFGs is that often the correct parse tree receives a slightly lower probability than an incorrect one. Charniak and Johnson (2005) use a maximum

entropy ranker to determine the best parse tree among the n-best possibilities provided by the generative model of Charniak (2000). *Discriminative parsing* is another approach that represents the task as a series of classification problems. Ratnaparkhi (1997) proposes a bottom-up and left-to-right approach, using a maximum entropy classifier to make decisions for constructing individual phrases. More recently, research has started to use conditional random fields to model the whole tree structure instead of only parts of it (Finkel et al., 2008; Tsuruoka et al., 2009).



(a) constituency tree (b) dependency tree

Figure 2.3: Sample parse trees generated using Stanford's NLP toolkit trained on the Penn Treebank.

2.4.4.2 *Dependency Parsing*

Syntactic dependency parsing is the task of deriving a parse tree based on binary head-dependent relations between the each word in the input sentence (Jason Eisner, 2007). The resulting tree is a graph where the nodes are the words, the edges are the binary dependency relations, and the finite verb is linked to a ROOT node (see Figure 2.3b). Even though the parsing is not centred around phrase structure, this is still represented though implicitly since "the head of a phrase comprises the whole phrase" (Covington, 2001). Additionally, dependency parsers assign a label to each edge, which represents the type of the dependency relation between the head and the dependent.

Yamada and Matsumoto (2003) suggested a left-to-right transition-based dependency parsing algorithm with three actions — *shift*, *right*, and *left* — based on a discriminative machine learning approach using a support vector machine classifier with context features. Another algorithm with an additional *reduce* action was suggested by Nivre (2003). Both

studies were later used as a foundation for the MaltParser (Nivre et al., 2006) — one of the most commonly used dependency parser implementations. One of the difficult aspects of this approach is the design and optimisation of features for the machine learning algorithm. An interesting innovation in this respect is presented by Chen and Manning (2014) who employ artificial neural networks to generate the feature vectors instead of designing them by hand. Another interesting approach involving transition-based dependency parsing attempts to perform the task simultaneously with POS tagging (Bohnet and Nivre, 2012).

Another notable approach to dependency parsing treats the task as a graph problem rather than computing a sequence of transitions. The main idea is that the task can be formalized as the search for a maximum spanning tree in a directed graph (McDonald et al., 2005). Some notable implementations of the graph-based approach are the MSTParser described by McDonald and Pereira (2006), and later the Tools for Natural Language Analysis, which uses a significantly faster implementation described by Bohnet (2010).

2.4.4.3 *Chunking*

Chunking is the task of identifying non-recursive phrases in text (Abney, 1991, 1995). It can be regarded as a kind of "shallow" parsing since it does not produce a fully hierarchical phrase structure, which makes it a less challenging task than dependency or constituency parsing. This can make it a preferable choice for syntactic analysis in applications involving non-standard language, such as clinical text. Even though chunking does not provide as much syntactic information as full parsing, it is an excellent method for identifying base noun phrases, which have a key role in higher level tasks such as named entity recognition.

Even though there have been some chunking systems using rule-based approaches (Grover and Tobin, 2006; Vilain and Day, 2000), the predominant method for automatic chunking is using statistical sequential tagging methods similar to statistical POS tagging (Tjong Kim Sang and Buchholz, 2000; Kudo and Matsumoto, 2001; Sha and Pereira, 2003). Since chunks typically comprise more than one token, the chunk tagset usually defines the beginning, the inside (or ending), and the outside tags using a BIO scheme (Tjong Kim Sang and Buchholz, 2000), standing for *beginning*, *inside*, *outside*. There is also an alternative scheme, which additionally distinguishes between *ending* tokens, and *single* token chunks, producing a finer representation of the chunk tags (Kudo and Matsumoto, 2001).

One of the great advantages of chunking is the simplification of a fairly complex problem to a classification-based task, which allows it to be solved using a wide variety of machine learning methods. Generally, chunking can be approached as a left-to-right (or the opposite) processing task using standard classifiers such as SVM (Kudo and Matsumoto, 2001) or MaxEnt (Koeling, 2000), or as a sequence labelling task using a structured classifier such as a CRF (Sha and Pereira, 2003) or maximum margin Markov networks (Buzhou et al., 2008).

2.4.5 *Word Sense Disambiguation*

Natural languages are ambiguous, many words have multiple possible interpretations. Word sense disambiguation (WSD) is the task of selecting the intended meaning of a word in a particular context. It can be applied in two ways depending on the words subjected to disambiguation (Navigli, 2009). The word meanings that are the end point of disambiguation are typically sourced from structured language resources, such as machine-readable dictionaries (Proctor, Paul, 1978; Soanes, Catherine and Stevenson, Angus, 2003), thesauri (Kilgarriff and Yallop, 2000), and ontologies (Miller et al., 1990; Fellbaum, 2005), but they can also be derived from unstructured resources such as corpora and word lists.

Essentially, WSD can be represented as choosing a single meaning out of a closed set, which is roughly the definition of a classification task. Supervised machine learning methods have used a range of different classifiers: Naïve Bayes (Singh et al., 2014), MaxEnt (Suárez and Palomar, 2002; Dang and Palmer, 2002), decision trees (Pedersen, 2001, 2002), SVMs (Buscaldi et al., 2006), and CRFs (Hatori et al., 2008). The main drawback of supervised learning approaches is the knowledge acquisition bottleneck (Gale et al., 993), since the training data has to contain examples of each word annotated with its intended meaning in many representative contexts. There are a number of methods used to automatically expand the available annotated data, such as using monosemous related words (Leacock et al., 1998; Mihalcea and Moldovan, 1999; Agirre and Martínez, 2004), parallel corpora or text translations (Diab and Resnik, 2002; Ng et al., 2003; Wang and Carroll, 2005; Wang and Martinez, 2006), the hyperlink graph of Wikipedia (Mihalcea, 2007), or distributional semantics thesauri (Miller, Tristan and Biemann, Chris and Zesch, Torsten and Gurevych, Iryna, 2012).

Unsupervised WSD approaches try to address the knowledge bottleneck issue. The corpus-based unsupervised approach used by Schütze (1998) essentially clusters human annotated words based on their contexts, and then projects new occurrences into the same vector space and assigns them to the cluster with nearest centroid (the average projection of its members). Pantel and Lin (2002) propose another clustering approach based on a similarity measure using dependency relation pairs. Mccarthy et al. (2004) describe a corpus-based approach, which integrates information from a manually constructed ontology.

Finally, there are knowledge-driven methods, which rely on different language resources (mentioned above) to approach the task (Lesk, 1986; Galley and McKeown, 2003; Navigli and Velardi, 2005). These methods typically have wider coverage than their supervised machine learning counterparts, due to the large scale of the resources they use, but are generally less accurate (Navigli, 2009).

2.4.6 *Information Extraction*

Information extraction is the automated extraction of structured information from unstructured data sources, typically, but not restricted to natural language text. It differs from information retrieval (IR) in the sense that the latter returns documents that satisfy a query (e.g. search engines), rather than structured information (Meystre et al., 2008). Due to the broad definition of IE, researchers have taken specialised approaches to different aspects of the problem, rather than come up with a single universal approach. Therefore, information extraction can be considered an umbrella term for a number of tasks with similar rationale and methodology, but with different aims and specific solutions.

Although there are variations, the most common tasks in information extraction aim at recognising entities (expressed using single words or phrases), references to and relations between them, and expressed events. *Relation extraction*, the recognition of particular relations between known entities in the text, was one of the earliest developments in the field. It can be simplified to a variation of pattern matching based on world knowledge and the linguistic analysis of the text produced by NLP processes such as part-of-speech tagging and parsing. The JASPER system (Andersen et al., 1992) and the NAS system

(Kuhns, 1988) are examples of early systems using this method to process news wire stories, and extract structured information about company mergers and acquisitions.

In these instances, the entities to be recognised were given to the system, and their attributes had a specific form such as the price per share in dollars, or the year quarter number of a report. To be able to generalise such systems required a way to recognise and classify entities such as company names, names of executives, numeric values, and others (referred to as named entities), which led to the definition of the *named entity recognition and classification*[8] task in MUC-6 (Grishman and Sundheim, 1996). Supervised learning approaches are currently the predominant group of methods used for NER, although some studies have used semi-supervised and unsupervised methods. Brin (1999) proposes a semi-supervised method that uses identifiable "seed" entities to extend the range of contexts recognised by a system based on regular expression matching. For example, book names and authors are likely to be mentioned in the same style on the same web page, e.g. *The Lord of The Rings, by J.R.R. Tolkien* and *Of Mice and Men, by John Steinbeck*, so a rule can be inferred from the first of these to be able to detect the second. Shinyama and Sekine (2004) propose an unsupervised method for discovering and classifying new entities through analysing the distribution of rare words in articles for a given time period, assuming that named entities would have different distributions to normal nouns, and thereby identifying and classifying new named entities from huge amounts of unannotated text. The CoNLL-2003 shared task introduced a corpus based on the Penn Treebank annotated with named entities (Tjong Kim Sang and De Meulder, 2003), which is still a commonly used benchmark for NER systems. The shared task is also a good example of NER studies using supervised learning approaches, as the vast majority of the participants used machine learning methods. Balasuriya et al. (2009) automatically inferred named entity gold standard annotation from Wikipedia's hyperlink structure, creating a vast new language resource. The authors also show that models trained on that corpus outperform ones trained on the Penn Treebank models when applied to Wikipedia text by a margin of 7.7 percentage points. In recent years, NER research has also been driven forward by domain specific studies, such as in clinical, chemical, and biomedical domains (A. Roberts and R. Gaizauskas and M. Hepple and Y. Guo, 2008; Corbett and Copestake, 2008; Alex et al., 2007).

8 The task is more commonly referred to as *Named Entity Recognition* or NER, including in this thesis.

Additionally, the subtask of named entity disambiguation has emerged, as the related knowledge bases of named entities have grown.This subtask, determining the real-world referent of an entity, has become more difficult, as the field achieves a real-world scale. There is often more than one entity instance of the same type. For example, there are sixteen different places that Wikipedia refers to on its disambiguation page for the name *Norfolk*, ten of them being in the USA[9]. Alhelbawy and Gaizauskas (2014) use a three step process based on graph ranking to disambiguate between such collections of named entities.

Another task that arises in relation to named entity recognition and information extraction (also originating from MUC-6) is anaphora resolution, often referred to in IE as coreference resolution. In short, the task can be defined as the identification of the parts of text that refer to the same discourse entity (Poesio et al., 2011)[10] as illustrated in Example 2.2.

> *Apple announced it will release the company's new iPhone on the market when the device is ready.*

Example 2.2: Pieces of text coreferring to *Apple Inc.* and *iPhone*.

Poesio et al. (2011) give an extensive account of the historical development of coreference resolution, as well as some of the more recent trends. The majority of modern methods approach the problem by constructing coreference chains that link together parts of the text that refer to the same entity using machine learning classifiers, and breaking down the task into recognising candidates and finding their coreference chain or starting a new one. This approach was first suggested by Soon et al. (2001), and is now commonly referred to as the pairwise coreference model. Although many recent systems are based on that approach, the most recent best accuracies were achieved by rule-based systems like the Stanford multi-pass sieve algorithm (Lee et al., 2011), and by unsupervised multigraph clustering, which achieved comparable results (Martschat, 2013).

9 `https://en.wikipedia.org/wiki/Norfolk_(disambiguation)`, *last accessed on 14th November 2015*

10 Van Deemter and Kibble (2000) discuss differences between the two terms, anaphora and coreference, that exist in certain contexts. However, Poesio et al. (2011) explain why the two terms can be used interchangeably in the context of this thesis.

The target domain has a great influence on the performance of both rule-based and machine learning NLP approaches. Applying a given model or algorithm to text with different properties usually results in lower performance than was achieved on the text used for development. The errors could be caused by unknown words and constructions, or by the inability of the approach to generalise sufficiently beyond the data used in development. In either case, a transition to a different domain often requires some adaptation of a model or algorithm. Additionally, not all domains have the same challenges in terms of language analysis, thus the goals of the NLP tasks may vary.

This section presents a range of approaches for analysing clinical text. Ensuring that NLP preprocessing is appropriate for the data is especially important, as errors could easily propagate to subsequent processes. In the context of information extraction, segmentation, word sense disambiguation, part-of-speech tagging, and parsing can be considered as preprocessing steps. Section 2.5.1 presents several studies which have adapted part-of-speech tagging and both constituency and dependency parsing to the clinical domain. This is followed in Section 2.5.2 by a discussion of common language-based application tasks involving analysis of clinical text.

2.5.1 *Preprocessing*

Information extraction, as previously described, usually relies on a set of text preprocessing steps. Hobbs (1993, 2002) lists the components of a typical IE system as being a tokenizer, sentence boundary detector, part-of-speech tagger, morphological analyzer, shallow parser, deep parser (optional), gazetteer (lists of location names), named entity recognizer, discourse module, template extractor, and template combiner. The first six of these components can be considered to be preprocessing steps, which would generally need to be specially developed or adapted for application to clinical text.

Spelling correction is an example of a task that has different implications depending on the exact type of clinical text it is applied to. Tolentino et al. (2007) describe a system for spelling correction of reports of adverse events following immunization, based on dictio-

naries, in their case the Unified Medical Language System (UMLS), and an edit distance metric. Even though it is generally perceived as a preprocessing step meant to enhance the quality of higher level tasks, sometimes the direction can be reversed as in the case of the work of Ruch et al. (2003), where WSD and NER modules are used to aid an edit distance spelling module. However, it is unclear if these techniques are applicable to primary care text, as they are clearly targeted at reports written in a style closer to standard language. Applying an automatic spelling correction mechanism to GP notes may result in more harm than gain. For example, the abbreviation *re* could be reasonably interpreted as *regarding*, *reply*, or *recommend* (the latter being the most common meaning in GP notes), but it is far more likely to be matched with *red* by any spelling module using edit distance metrics or even part-of-speech tags, as the abbreviation is usually followed by a phrase like *fybogel twice a day*.

Part-of-speech tagging has been previously applied to text in specialised domains like biomedical literature and tweets. Both the GENIA tagger (Tsuruoka et al., 2005) and the ARK tagger (Gimpel et al., 2011) show that in-domain training data is critical for the good performance of a statistical model. The GENIA tagger is an especially good example, because it shows that adding out of domain data, such as the Wall Street Journal part of the Penn Treebank slightly decreases accuracy. Similar results involving clinical text are reported by Coden et al. (2005), who compare POS taggers using the Penn Treebank, GENIA, and a small part of the MED corpus annotated with parts of speech (Pakhomov et al., 2004) as training sets, and evaluating the models on clinical text. However, Fan et al. (2011) describe an experiment which involved two clinical data sets of progress notes from different institutions, annotated with POS tags. The study showed that the sources of training and evaluation data can influence the model performance considerably, although in all cases it remained higher than the performance of the cTAKES POS tagging module (Savova et al., 2010). In a different approach Ferraro et al. (2013) show that using a small amount of annotated in-domain data can improve the performance of POS tagging models trained on standard text by using domain adaptation algorithms (Daumé et al., 2010).

Word sense disambiguation in the clinical domain has been largely focused on expansion of abbreviations and acronyms, which can be seen as a slightly simpler WSD task. Moon et al. (2015) provide an overview of the problem, discussing the language and privacy issues of the data along with the lack of language resources with clinical abbreviations

and acronyms, or even comprehensive lists of such entities and possible "expansions" — Xu et al. (2009) and Joshi et al. (2006a) being notable exceptions. The problem is frequently approached using a discriminative machine learning methods with various classifiers (Pakhomov, 2002; Joshi et al., 2006a,b; Stevenson et al., 2009; Moon et al., 2015). While the majority of classification approaches use standard feature sets such as bag of words, POS tags and other linguistic features, and semantic information from UMLS concepts and location in the text structure, some studies have also employed word representation features. Wu et al. (2015) and Li et al. (2015) suggest using different modifications of the embeddings proposed by Mikolov et al. (2013) to improve the feature sets of the WSD classification models. In contrast to the abbreviation and acronym expansion studies, Savova et al. (2008b) explore discriminative WSD methods in biomedical and clinical texts focusing on the contribution of different feature types.

Syntactic level processing parsing is essential for capturing chunks of valuable medical terminology in medical and clinical texts, so parsing has been a steady topic of research since the 1970s (Hirschman et al., 1976; Sager et al., 1987; A. M. Rassinoux and R. H. Baud and J. R. Scherrer, 1994; Baud et al., 1998). Early approaches used dictionaries and sets of rules and constraints to parse clinical text, while machine learning methods have been developed more recently, based on the necessary in-domain annotated training data. Xu et al. (2011) reported 81.0 f-score using the Stanford constituency parser on a small manually annotated part of the i2b2-2010 dataset, which shows a significantly lower performance than general edited text. Fan et al. (2013) reports an almost equal result (81.1) for their dependency parsing models trained using the Stanford parser and a combination of clinical and newswire text data. Finally, Jiang et al. (2014) compares three prominent parser implementations on two clinical treebank resources (Albright et al., 2013; Fan et al., 2013), and the Penn Treebank, concluding that the highest accuracy is achieved by training on a combination of general domain and domain-specific dependency treebanks.

2.5.2 *Information Extraction*

Clinical NLP has tackled a wide range of problems using many different approaches. However, most can be regarded as focusing on one or more of the same three basic information

extraction tasks discussed in Section 2.4.6: recognising concepts, finding and linking references to concepts (i.e. coreference resolution) and identifying relations between concepts.

Clinical NLP often faces difficult information access issues due to the sensitive content of the data. Text de-identification is one of the necessary supporting tasks when working with clinical data. The task aims to remove identifying information from clinical data in order to prevent immediate connection to patients' identity. Automating this task is an important step towards better data access for the community. Although criteria vary across countries of origin, clinical data is generally protected by privacy laws that severely restrict access to it. Ideally, the data should be anonymised preventing any link to the patients, but without causing qualitative change in the data (Meystre et al., 2014; Walker, 2015). Bodies responsible for information governance commonly define a set of concepts, called protected health information (PHI), to be pseudonymised or redacted before wider access to the data is granted. From a technical point of view the process is similar to NER and concept recognition, as the targeted entities are typically names of patients, clinicians and facilities, addresses, and dates of birth. The main challenges of the task are its virtually zero error tolerance, and high human annotation cost (Dorr et al., 2006).

Historically, a number of studies have suggested automated de-identification that achieved f-score results in the mid- and upper 90s using various methods including customised algorithms for each PHI type (Sweeney, 1996), regular expressions (Fielstein et al., 2004), machine learning (Taira et al., 2002), and even adapting a WSD system (Ruch et al., 2000). Sibanda and Uzuner (2006) created surrogate PHIs put in place of the real redacted PHIs, and then used an SVM classifier to recognise them, virtually simulating the process of de-identification. The same corpus with more realistic surrogates was used for the de-identification challenge of the 2006 i2b2 shared task (Uzuner et al., 2007b). The best performance among the participants, 98.35% f-score, was achieved by Wellner et al. (2007), using a CRF entity recognition model. Track 1 of the i2b2 challenge from 2014 (Stubbs et al., 2015a) sought to improve the anonymisation of longitudinal patient records by removing even more PHI than required under the US Health Insurance Portability and Accountability Act (HIPAA). The overall accuracy was lower than the previous task as the complexity had increased, but the highest performances still yielded f-scores in the 90s with a top score of 93.6 achieved by Yang and Garibaldi (2015).

Electronic health records often contain information about patient attributes such as smoker status, or presence of a particular disease or condition. The i2b2 challenges have included tasks to identify these kinds of attributes. One of the tracks in 2006 sought to recognise patient smoker status (Uzuner et al., 2007a), while the challenge in 2008 aimed to determine if the patient is obese before determining a set of co-morbidities associated with that condition (Uzuner, 2009). The majority of participants used machine learning based methods although the problems were approached in different ways (Aramaki et al., 2006; Carrero et al., 2006; Cohen, 2008). Some methods were rule-based (Guillen, 2006), and some used pre-existing IE systems (Heinze et al., 2008).

The i2b2 challenges have included clinical concept recognition tasks several times, focusing on identifying obesity comorbidities (Uzuner, 2009), medication (Uzuner et al., 2010a), and risk factors for heart disease (Stubbs et al., 2015b). Since clinical concepts are expressed via medical terminology, lexical resources such as UMLS can play an important role in their recognition. Generally the participants in all challenges favoured rule-based approaches, but there is an increasing trend towards using a machine learning element in combination with rules and terminology resources. The rule-based approaches mostly rely on lexical resources to detect candidate terms and rules to determine their validity, e.g. recognising negation using NegEx (Chapman et al., 2001). On the other hand, machine learning approaches used mainly CRF and SVM classification (Savova et al., 2008a; Patrick and Li, 2009; Halgrim et al., 2010; Chen et al., 2014; Roberts et al., 2015) to recognise entities, but still made use of rule-based tools like NegEx or heuristics for boosting their results through pre- and post processing steps. Wang and Patrick (2009) suggested a two-tier classification method aimed at resolving correctly recognised, but incorrectly classified clinical concepts. SVM and MaxEnt classifiers were used to re-classify the concepts that were initially recognised by a CRF classifier. The final class of each concept was decided through a weighted voting process between the classifiers, which led to an f-score increase of 3.35 points.

Detailed information is at the heart of clinical data, but to access it, a system needs to be able to identify certain relations and attributes of the targeted medical concepts (entities). Some examples are detecting negation, determining absolute and relative time references, as well as relations between medical concepts such as disease indication or causation. R. Gaizauskas and H. Harkema and M. Hepple and A. Setzer (2006) suggested that entities

and relations from a patient's EHR should be integrated into a chronicle covering their condition, diagnosis and treatment over the period of care. They proposed an IE system containing pre-existing tools within the GATE processing platform (Cunningham et al., 2002) using a rule-based algorithm for relation extraction. More than 332,000 clinical narratives about roughly 37,000 cancer patients were used in their study. The system concentrated on a limited number of temporal relations achieving 72.83 precision and 58.70 recall. A more sophisticated approach was suggested by A. Roberts and R. Gaizauskas and M. Hepple (2008) who treated the relation recognition problem as a classification task, and designed a GATE pipeline that used an SVM classifier module for relation recognition. As the classification process should consider all pairs of candidate entities for possible relations, the pipeline also made use of some heuristics in order to decrease the number of pairs considered. The approach was applied to the full clinical records of more than 20,000 cancer patients from the Royal Marsden Hospital.

Two of the i2b2 challenges also focused on extracting relations from clinical text. The 2010 challenge investigated groups of relations regarding treatments, tests, and medical problems (Uzuner et al., 2011). The participants were given access to 394 reports (progress reports and discharge summaries) for training, and 477 for testing, plus 877 unannotated reports. Most participants used SVM classifiers with the best performance reaching 73.7 f-score. The challenge organisers concluded that the relatively low performance was due to the lack of explicit contextual information to determine relations, and the complexity of the language. The 2012 challenge focused on temporal relations in addition to clinically significant events and temporal expressions (Sun et al., 2013b). The data used for the challenge comprised 310 discharge summaries from Partners Healthcare and the Beth Israel Deaconess Medical Center. The systems that used hybrid approaches combining machine learning and heuristics achieved the highest performance. However, the highest performing system achieved only 69.00 f-score, which hints at the unresolved challenges facing the task of clinical temporal reasoning (Sun et al., 2013c).

Despite its importance in IE applications in other domains, coreference resolution is, as Zheng et al. (2011) points out, one of the less explored areas of clinical NLP. The first annotated data resource in this area was created during the 2011 i2b2 challenge (Uzuner et al., 2012), and to date it remains the most significant resource of its kind. The challenge provided data from the Ontology Development and Information Extraction Corpus

in addition to data from the i2b2 VA corpus (data from Beth Israel Deaconess Medical Center, Partners Healthcare, and University of Pittsburgh Medical Center), amounting to 978 reports of various types. The challenge consisted of three tasks. The first task consisted of identifying concept mentions in the text, and performing coreference resolution on the recognised mentions to construct coreference chains. The second and third tasks focused only on coreference resolution using text with ground truth concept mention annotations on the ODIE corpus and the i2b2/VA clinical records respectively. The tasks were approached using rule-based, statistical, and hybrid approaches, which often complemented each other with regard to their errors. The top f-scores of the participating systems were 82.4, 91.5, and 91.4 for the respective tasks. Considering this relatively high performance, Uzuner et al. (2012) concluded that the systems perform well, but face difficulties in solving coreference in cases requiring domain knowledge. They point out that the key to improving performance is further integration of domain knowledge. More recently, Jindal and Roth (2013) have made use of the i2b2 coreference corpus, and successfully applied constraints for improved pronoun resolution. Jindal et al. (2014) have built upon the idea and used it in a new approach that replaces the common pipeline approach of linking mention pairs through a sequence of inference steps, with one joint global inference process. It should be noted that both ideas are domain-independent and could be applied to coreference resolution of text in other domains.

3

BUILDING THE HARVEY CORPUS

Applying a natural language processing (NLP) system to text in a genre or domain that is different to the text used for its development is still one of the greatest challenges in NLP. Retraining or redeveloping the system using a language resource representative of the new, target domain is currently the safest approach to addressing this problem. This chapter describes the building of the Harvey Corpus to support one of the main goals of this thesis — producing a medical concept extraction system for primary care text. It provides a detailed account of its preparation, assembly, and evaluation. The corpus was built based on two principles: 1. randomly selected data of the targeted type to ensure appropriate representativeness, and 2. reusable annotation reflecting the properties of the target language, while serving the final goal of the thesis. While a number of technical difficulties emerged throughout the whole process, the main challenges of critical importance for the final result were the annotator training, and the balance between the complexity and informativeness of the annotation guidelines.

Section 3.1 gives information about the origin of the target data, and reviews the peculiar characteristics that make it challenging to process with existing NLP tools. Section 3.2 recounts the rounds of annotation guideline development, and the document that was produced as a result[1]. Section 3.3 traces the stages of assembly of the Harvey Corpus, starting from the selection and annotation of the data, to its marshalling as a data structure, and current availability. Finally, Section 3.4 describes an extrinsic evaluation of the corpus to establish whether the annotation quality is sufficient to support a stable training process for a statistical model derived from the corpus.

1 The final version of the annotation guidelines discussed in this chapter is available in Appendix A

3.1 GPRD DATA

The Harvey Corpus was built from a subset of free text GP notes obtained under a licence for the purposes of the Patient Records Enhancement Programme (PREP). This research project was funded by the Wellcome Trust in 2008 "The Ergonomics of Electronic Patient Records" (Grant No. 086105/Z/08/Z). The project addressed the potential of free text to augment structured primary care electronic health records, through interdisciplinary work in areas including epidemiology, human-computer interaction, and natural language processing.

The data was manually de-identified by the GPRD before it was acquired for PREP. All protected health information (PHI) in the text (personal names, dates of birth, phone numbers, and addresses) was replaced with strings of tilde characters of the same length as the original string.

The pool of notes that the corpus data was selected from can be divided into three major categories depending on content: *letters to and from specialists*; *test and scan results*; and *general notes of a patient visit or interaction* (see Figure 1.1 in Chapter 1). The letters are usually very descriptive and detailed, grammatically well written, and generally meant to clearly communicate a message between the GP and a specialist. The test and scan results primarily contain result values, and optionally additional comments on the results. The general notes are about various kinds of patient interactions – telephone encounters, home visits, hospitalisation, etc. – but mostly they are about interaction with patients at a general practice. The last kind of notes are often divided into a part that describes what the patients said about their problems, and a part that records the GP's train of thought during examination, which might variously include observations, conclusions, reflection on alternatives, and proposed further action. The two parts are commonly separated by a phrase or an acronym that roughly means "on examination", e.g. *o/e*.

The general notes, as illustrated in Example 1.1 in Chapter 1, are written in a sub-language characterised by extreme brevity and a telegraphic style of expression. The quality and presence of punctuation varies from completely missing to well placed commas and end of sentence markers. There is also an abundance of spelling mistakes due to mistyping or omitted spaces between words.

Another important feature of the free text notes is the use of heavily abbreviated words and terminology, often trimmed down to a single letter. GPs also use a number of signs and short jargon words to denote commonly used longer words or expressions. However, the issue with possibly the greatest impact on language processing is the lack of standard grammar as clinicians strive for terse expression.

A free text note is often a list of items that the patient talked about, or the GP observed or looked for during the examination. Thus, many function words are omitted and the text is presented in a virtually list-like format of items of interest and their respective pertaining information. In fact, the use of the copula is so rare that it could be perceived as an exception rather than the grammatical norm.

Some of the characteristics described above are also observed with other types of text, however the combination of them together greatly intensifies the need for the reader to have context awareness and domain knowledge to fully comprehend the text. Consequently there are a number of specific obstacles before it is possible to carry out any automatic processing of the data using NLP technology, as well as before any manual processing by experts without medical training.

Since the text notes often begin mid-sentence assuming the Read term (the text representation of a Read code) as part of the text, it made sense to include the Read term in the text. However, the term and the rest of the note were separated by a double pipe sign (||) in order to keep the distinction.

3.2 ANNOTATION DESIGN

When developing a new annotated corpus, one of the key decisions is whether to adopt an existing annotation scheme and guidelines or to design new ones. Even though the CoNLL-2000 shared task (Tjong Kim Sang and Buchholz, 2000) established a chunk annotated corpus standard, there are no written guidelines for its chunk annotation scheme. Picking a single annotation scheme for semantic entities also seems difficult, as even though there are quite a few annotated resources, they are usually quite specific and dependent on the task they were designed to support. Perhaps the only exception to this is TimeML (ISO, 2008), which was used in a number of studies as a basis for the scheme definitions and annotation guidelines for temporal events.

Another important issue is the choice of annotators and their background. Roberts et al. (2009) show that clinically trained annotators are potentially better at annotating clinical records with semantic relations than linguists or computer scientists. However, there is no clear evidence that this is true for linguistic annotation such as chunking. On the other hand, Fan et al. (2013) use linguist annotators for syntactic annotation of malformed POS-tagged sentences of clinical text. Ultimately the choice of annotators depends on the amount of effort and training that they would need to achieve comparable results. The intuition was that chunking should be relatively simple enough to teach to medical students with a basic understanding of grammar, while teaching linguists clinical vocabulary and basic background knowledge of clinical procedures seems like a more difficult task. Therefore the choice was made to train as annotators fourth year medical students with substantial medical knowledge and sufficient experience with GP notes. However, achieving good results depends also on keeping the annotation as simple and clear as possible to minimise the required linguistic training, and so a custom annotation scheme and guidelines needed to be devised for the Harvey Corpus.

A slightly more technical, but nonetheless important issue is the choice of annotation tool suitable for the task. The web-based annotation platform *brat* (Stenetorp et al., 2012a) was chosen, because of its clean and simple interface, flexibility, and centralised data storage. The platform allows remote access for the annotators, but originally did not prevent them from copying the text (the research license does not allow the re-distribution of the data). A small modification to the Brat source was implemented in order to achieve that. It also kept a log with a time stamp of all annotations, in order to roughly track the time periods the annotators were working for.

Finally, in order to access the progress of annotation design, an appropriate inter-annotator agreement metric needed to be chosen. Following previous practice in the field, the f-score was used as suggested by Hripcsak and Rothschild (2005), but anticipating the sparsity of the data, an alternative, more relaxed calculation is suggested in Section 3.2.1.

This section describes in detail the design and refinement of an annotation scheme and guidelines for chunking and entity annotation. They were developed in a similar fashion to the CLEF corpus and guidelines (Roberts et al., 2009) which adhere to the principles of language resource annotation for information retrieval formulated by Boisen et al. (2000). First, a draft version of the scheme and guidelines were developed (see Sections 3.2.2 and

3.2.3), and then incrementally refined with the helpful feedback of two medical students who subsequently became the first annotators (see Section 3.2.1). Finally, another medical student was trained to both annotate text and adjudicate the annotations of the other two (see Section 3.2.5).

3.2.1 *Inter-Annotator Agreement for the Harvey Corpus*

Calculating agreement between BIO-style annotations is typically done using an f-score as described in Section 2.2.4. The reason for calculating agreement between text spans is the very high number of possible annotation borders, even if borders can only be between tokens. One of the shortcomings of using the BIO f-score is that it is not forgiving towards partial matches. Given the sparsity of the data it was important to come up with a more relaxed metric.

The scoring definitions for the NER evaluation at the seventh Message Understanding Conference (MUC-7) are very similar to what was later suggested by Hripcsak and Rothschild (2005) for calculating inter-annotator agreement (see Section 2.2) — one annotator being assumed as the gold standard, and the other as the test output. While Hripcsak and Rothschild give a mathematical justification of why it is possible to use the f-score to calculate inter-annotator agreement, the MUC-7 scoring instructions give a better breakdown of how that should be implemented. Their approach gives perhaps the clearest perspective on calculating inter-annotator agreement, but with a small correction in how partial matches are treated. The guidelines take account of the possible need to include partial matches in the calculation, although they do not actually do that in their evaluation — that is they are either assumed to be part of the incorrect matches or discarded.

Five of the MUC-7 counting categories involved in the precision and recall calculation were used (see the full list of categories in Table C.1): CORRECT, INCORRECT, PARTIAL, SPURIOUS, and MISSING. To translate the meaning of these categories to inter-annotator agreement, one annotator is assumed to be the "evaluated system" and the other the "gold standard". Given that assumption, there are two aspects of an annotation that need to match the gold standard in order for it to be correct: the annotation category and the annotation word span. Naturally, annotations that match both are considered CORRECT, but not all of the rest are considered INCORRECT. Annotations that have the same annota-

tion category, but only partially matching word spans are considered PARTIAL only if one of the word spans fully contains the other as in the two different annotations of *the door* in Figure 3.1. Note that annotations which partially overlap with the gold standard are considered INCORRECT, in addition to annotations that do not match the category of their gold standard counterpart, for example, *city bus* and *bus driver* in Figure 3.1. Annotations by the "evaluated system" annotator which were not annotated by the "gold standard" annotator are considered SPURIOUS. Finally, the annotations which were made only by the gold standard annotator are considered MISSING.

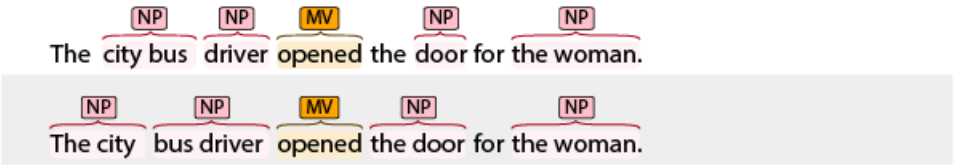


Figure 3.1: Two different annotations of the same text

Precision and recall are defined by the following two equations:

$$\text{precision} = \frac{\text{CORRECT}}{\text{COR} + \text{INC} + \text{PAR} + \text{SPU}} \tag{3.1a}$$

$$\text{recall} = \frac{\text{COR}}{\text{COR} + \text{INC} + \text{PAR} + \text{MIS}} \tag{3.1b}$$

Even though the definition of syntactic chunks can be strict about their boundaries, semantic entities such as symptoms often cannot be strictly defined, which leaves room for ambiguity and disagreement. Therefore, it is important to compute not only a *strict* conservative measure of agreement such as the ones defined in Equations 3.1, but also a flexible relaxed measure that acknowledges the cases where the annotators came close to fully agreeing. A second, *relaxed* form of precision and recall are calculated as well, treating the PARTIAL category as correct:

$$\text{precision}_\text{R} = \frac{\text{COR} + \text{PAR}}{\text{COR} + \text{INC} + \text{PAR} + \text{SPU}} \tag{3.2a}$$

$$\text{recall}_\text{R} = \frac{\text{COR} + \text{PAR}}{\text{COR} + \text{INC} + \text{PAR} + \text{MIS}} \tag{3.2b}$$

Another point in favour of having a relaxed measure is that the strict measure does not account for the way the final version of the data is produced. One can ignore the disagreement items, use a voting system in the case of more than two annotators, or use what is the common practice in the clinical NLP field, a third annotator (adjudicator) who ultimately resolves the cases of disagreement (Alnazzawi et al., 2014; Sun et al., 2013a; Uzuner, 2009). Using an adjudicator does not make the final data perfect, but it certainly improves its reliability. It is not clear how to fairly quantify that improvement, but if we have to pick the cases with the greatest chance of improvement, they would be the near matches of the PARTIAL category. Therefore it makes sense to present an agreement measure that accounts for that potential.

3.2.2 *Annotation Scheme*

The greatest challenge in the initial design of the annotation scheme was to find the appropriate balance between encoding enough information to support further research, and achieving clarity, simplicity, and conciseness in the guidelines. The annotation scheme had to capture as much syntactic structure as possible, while not "inventing" elements that were not there in order to create canonical structures. Adopting chunks as the main units of annotation was a logical solution, as Abney (1991) defines them as "the parse trees that are left behind after we have unattached problematic elements." In other words, chunking trades the levels of the parse tree closer to the root (the longer range relations) for better quality in the levels closer to the leaves (shorter range relations). But while chunking sacrifices information in standard grammatical text, it is appropriate for clinical notes because there is less tree structure to be lost.

Unfortunately there are few papers on chunking annotation. The only available comprehensive chunking guidelines seem to be those presented by Bharati et al. (2006); however, their design was targeted at Indian languages and annotators with a linguistic background, which made them unsuitable for the purposes of this thesis. A more popular approach to chunking, or shallow parsing, is the pruning of full parse trees, as suggested by Abney (1991). The CoNLL-2000 chunking challenge (Tjong Kim Sang and Buchholz, 2000) used this approach, trimming down a subset of the Penn Treebank (Marcus et al., 1993) to chunks using a pattern-matching rule-based script. Given the absence of similar prior

work a new annotation scheme was developed along with a set of corresponding annotation guidelines taking into account the telegraphic language style and many omitted words in the data. The background of the annotators was also taken into consideration, as they were expected to be native English speakers, but with limited understanding of linguistic theory and terminology such as parts of speech and syntax.

After preliminary discussions, an initial annotation scheme was produced and applied to a few records to enable any problems and possible improvements to be identified. The initial set of chunk types comprised noun phrase chunks (NPs), adjectival phrase chunks (APs), main verbs (MVs), and prepositional phrase chunks (PPs). Several alterations to the set of annotation types were made in order to make them clearer and simplify the task. Base noun phrase chunks were introduced because they allowed more flexible analysis than full noun phrases. Prepositional phrase chunks were excluded as many of them can be reliably recognised using pattern matching on top of NPs. The AP definition was altered to include only comparative expressions and predicative expressions such as *brown* and *better* in *My dog is brown* and *Patient's tummy feels better*.

On another note, producing language resources such as the Harvey Corpus requires significant amounts of money, time, and labour. This motivated looking for further useful annotation types that could be added to the scheme in order to make the annotation process more cost effective. Four additional types of semantic annotation, similar to what is commonly referred to as Named Entities (NE), were introduced as they were thought likely to be useful in future research. The following entity types were added: *quantitative expressions* (QE), *temporal expressions* (TE), *locative expressions* (LE) and *on examination expressions* (OE).

Quantity, frequency, and time of occurrence are important additional pieces of information not only for symptoms and diseases, but also for drug prescription and administration. Such information may contribute to symptom and disease recognition, but it is also useful for healthcare related research, such as studying drug side effects. *Quantitative expressions* cover all forms of the various quantities recorded in the data, such as *pulse 90*, *20ml*, etc. They should not be mistaken for identification numbers or any other numbers not signifying quantities. The only quantities that are not annotated as QEs are units of time, e.g. *1h*. *Temporal expressions* are defined as words, phrases or clauses that contain information related to time. They can manifest as a reference to a specific moment in time (*in two*

days), the duration of an event (*for two hours*), or an event's frequency (*twice a day*). Even though using TimeML for clinical text was popularised with the last i2b2 challenge (Sun et al., 2013a), using it for this annotation enterprise would have overcomplicated the annotation scheme given that the corpus would not contain any connections between the records. Location is also an important aspect of the information contained in clinical text. The location of the patient encounter (*home* vs. *clinic*) might be important, as well as the locus of a symptom (*joint pain*) or a disease (*lung cancer*). *Locative expressions* were introduced to mark these two types of location in the corpus. Finally, there are a number of expressions, such as *o/e*, that mark the border between patient narrative and the GP's train of thought, called *on examination expressions*. The ability to recognise such markers could provide contextual information. For example, speculative diagnoses before the marker are likely to be associated with the patient and after the marker with the GP.

Syntactic chunks and semantic entities were considered to be two separate almost independent groups of annotations, which were bound to co-occur in some cases. Therefore, a set of rules governing such co-occurrences needed to be established. The following rule was introduced to ensure that no annotation embedding was done within the same tagset:

1. **Rule of structure simplicity**: no chunk annotation can be embedded in another chunk annotation, and no semantic entity annotation can be embedded in another semantic entity annotation.

While the first rule does preclude embedding annotations of the same type (syntactic and semantic), it doesn't do so for annotations of different types. Additionally, it was assumed that all annotations should be representable as syntactic constituents, and therefore if their boundaries overlap, one of them must contain the other. If this is not the case, at least one of them surely is not a well-formed syntactic constituent. The following rule was introduced in order to reflect this assumption:

2. **Rule of compatibility:** annotation embedding may occur only when the annotation borders coincide or when one of the annotations is inside the other (inclusive border indices).

Figure 3.2 illustrates correct and incorrect use of embedded annotations according to the rules defined above. The first sentence illustrates a contradiction to the rule of simplicity:

an AP is embedded in an NP, and a QE in a TE. The embedded annotations of the second sentence partially overlap each other without any of them fully containing another annotation. The annotations in the third sentence show the correct way of embedding, complying with both rules.

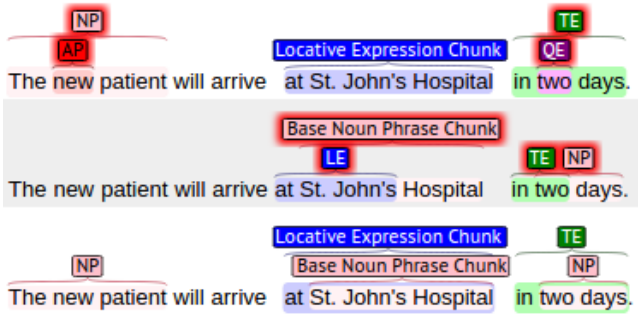


Figure 3.2: Examples illustrating correct (line three) and incorrect (lines one and two) use of embedded annotations.

3.2.3 *Annotation Guidelines*

A document was developed (available in Appendix A) to describe the annotation types and to explain how difficult cases should be treated, to ensure consistency. The goal was to write the document as a training manual, while including enough examples so that it could be used as a reference during annotation. It was meant to address the expected lack of linguistic knowledge of the annotators by giving a short practical introduction to English grammar[2].

The guidelines cover three main topics. After a short introduction to the project goals and expectations, the first part introduces the reader to the basics of grammar. It describes the concepts of phrases and parts of speech, concentrating on verbs, NPs, and APs in particular. The main purpose of this section is to define the basic concepts used in the rest of the guidelines thus allowing the training of annotators without any linguistic background. The second part of the guidelines provides detailed definitions of the annotation chunks and expressions, along with examples and special cases that can be used as a quick

2 Some of the linguistic theory and explanations were simplified in order to make them more accessible to annotators without a linguistic background; as a result the explanations do not completely comply with conventional linguistic theory.

reference manual during annotation. The last part of the guidelines helps to increase the quality and consistency of the produced annotation by giving practical advice on some common issues and detailed instructions on how to handle particular situations — they urge the annotators to be confident in their opinion, while not annotating text they do not understand. The annotators are also encouraged to consider the possible content of redacted text in their analysis, and to annotate acronyms and abbreviations whenever they can be identified as chunks or expressions. Key issues such as punctuation, conjunctions, and embedding of annotation are also discussed in the final part of the guidelines.

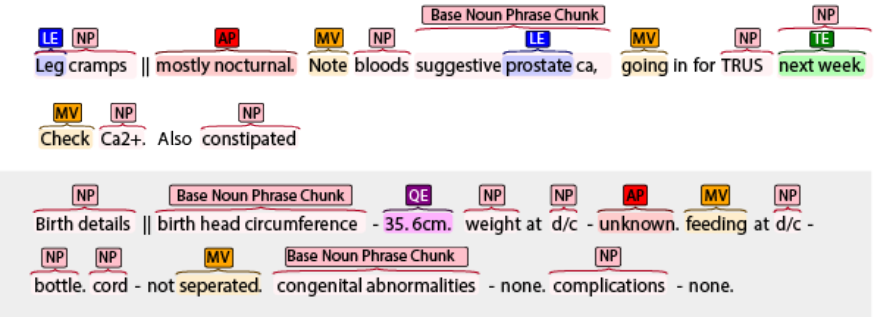


Figure 3.3: Brat annotation showing labelled spans

The guidelines also include a short introduction to the Brat annotation platform. Brat allows the annotators to work with a web-based interface from a remote location (see Figure 3.3 for a screenshot of part of the annotation window), while preventing them from downloading any of the data. Finally, the guidelines describe the adjudication process and the role of the third annotator, which follows the example of Roberts et al. (2008) in restricting their duties to resolving annotation conflicts without adding or removing any information. The annotators are considered to agree when both of them have provided the same borders and tag for an annotation. In cases where only one annotation has been provided, it is considered to be correct as it is the only one available. The judge should intervene only in cases where candidate annotations overlap, using their own judgement to select the better annotation.

3.2.4 *Refinement*

After the guideline development and refinement process, following Roberts et al. (2008), an iterative process was set up going through annotation, evaluation, and refinement stages. The plan was to send out small batches of 25 to 50 records to the annotators and analyse their results to improve the guidelines to a sufficient level. The aim was to create a set of guidelines that would allow anyone to learn and produce reasonable quality annotations with minimal in-person training. Such training was avoided initially in favour of independent self-training, because it was believed that teaching by example might prevent the annotators from learning the appropriate linguistic generalisations.

The two domain experts (referred to below as annotators A and B) annotated fifty records remotely over the course of two weeks during the first annotation round. The agreement achieved only 35 f-score, which is the lowest that was ever measured throughout the experiments. An error analysis identified a few basic problems with the guidelines, including an ambiguity in the definitions of NPs and APs, which led to a great number of errors as they comprise a dominant part of the annotations. The two annotation types needed to be made more clearly distinguishable from one another. At this point the basic grammar section was simplified, a definition of gerunds was added, and on-examination expressions were clearly redefined as markers between sections. The error analysis conclusions were also confirmed by feedback from the annotators. They suggested that the examples in the guidelines should be improved and expanded. This prompted the creation of an interactive tutorial using the Brat platform to show definitions of all annotations with made-up examples, while asking the annotators to test their skills and compare them to a solution key. During this first refinement round very little was changed regarding the definitions of semantic entity annotations. The annotators did not feel confident in creating embedded annotations, and so annotated semantic entities only sporadically, which resulted in extremely low agreement in that category.

The updated guidelines led to significantly better results in the second annotation batch. The agreement in all chunk categories and the on-examination expression improved, as well as overall agreement f-score, which reached 43. However, there were considerably more instances of the other expression annotations, which decreased agreement in those specific categories even more.

In order to gather more feedback from the annotators, a workshop on the use of the guidelines was organised before the second refinement stage. The annotators were engaged in a series of discussions about each annotation type, stressing the relevant grammar points using non-medical examples and attempting to lead them to a correct understanding of the annotation through asking the right questions.

During the workshop it became obvious that the guidelines needed to explain the different roles of participles because the annotators experienced difficulty in distinguishing passive voice from adjectives, and continuous verb forms from gerunds. They also continued to avoid embedding different types of annotations, because the embedding rules were not clearly explained and illustrated by examples in the guidelines.

The third annotation batch had a steady overall improvement to 50 agreement f-score in all categories except APs. The APs continued to be a confusing concept for the annotators, so they were redefined to be as simple as possible, and an extensive range of examples was added. It was also noted that even though certain aspects of the annotation improved and became more consistent, others worsened significantly in a way that could not be attributed to an ambiguity or lack of information in the guidelines. This showed that there must be another reason behind the errors, or at least a big part of them. The Brat platform log showed that the annotators worked on small 5-10 record subsets at a time, with breaks of at least a day between them. This confirmed a suspicion that the annotators were not fully concentrating when doing parts of the annotation, which often made them inconsistent. It became clear that it would be difficult to preemptively list all possible wrong interpretations of the guidelines and adjust the guidelines accordingly or warn the annotators about them. Thus even though the IAA results were improving, a change of training approach was required. It was decided that the annotation scheme and guidelines had reached a stable level and any further efforts should focus on setting up a productive environment for the annotation process.

3.2.5 *Annotator Training*

The observations made during the first three annotation rounds suggested that the context of the annotation process could be just as important as the training instructions. The annotators had always been advised to work on as many records as possible in a single

session, but during the first three batches they did not follow that advice, which resulted in many short annotation sessions with low consistency. Another observation, made by the annotators themselves, suggested that their understanding of the annotation deteriorates over time, for example during the two-week gap between the second and third annotation batches. They also consistently found that the first few records in every session would take them more than the usual time and effort.

These issues were addressed through setting up the annotation sessions in a university computer lab rather than at home, with the author of this thesis present to answer questions, which were restricted to the interpretation of the guidelines and not about their application in a particular instance. The new setup aimed to increase annotator concentration, while also introducing some training into the process by making them generalise their questions in order to receive answers. A week before the fourth annotation round, a short tutorial was organised to refresh their skills and to address some of the error patterns from the previous annotation rounds. The new annotation strategy resulted in a jump in the overall agreement to 76 f-score, and a general increase in all separate categories, most notably in the chunks. Three out of the next four annotation sessions yielded similar results within 5 base points (see Figure 3.4), which indicated that the annotators had achieved a sufficient level of consistency to start producing annotation for the corpus.

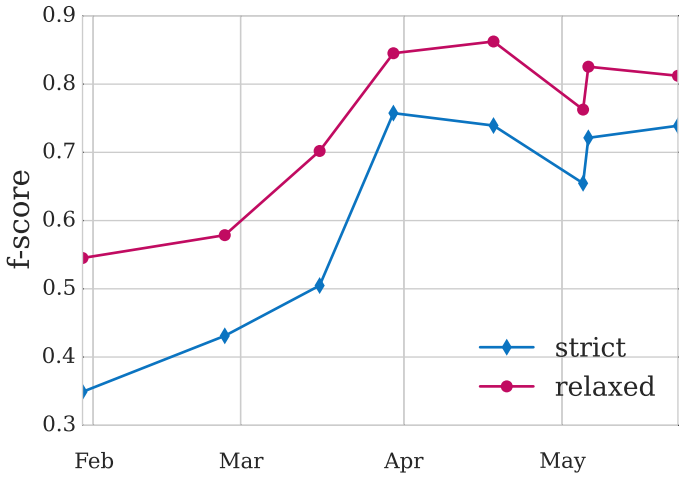


Figure 3.4: Inter-annotator agreement during the training period

The training of the third annotator (referred to below as annotator C) started when annotators A and B had almost completed their training. The selected domain expert was

given a short introduction to the project and the guidelines before being assigned the first annotation batch. The annotation quality of the first batch appeared encouraging although it was hard to evaluate it using IAA as the annotation quality of the other annotators was low at the time.

After the first batch, annotator C was given two more annotation rounds with feedback and took part in a workshop along with the other annotators at the end of the training phase.

	Strict			**Relaxed**		
	Precision	**Recall**	**f-score**	**Precision**	**Recall**	**f-score**
chunks	0.65	0.64	65	0.82	0.80	81
entities	0.50	0.56	53	0.69	0.78	73
all	**0.57**	**0.57**	**57**	**0.71**	**0.71**	**71**

Table 3.1: IAA between annotators C and D on their training annotation batches. The results in *all* are calculated as microaverages.

Unfortunately annotator A exited the project before its completion for personal reasons and due to scheduling issues was replaced by annotator C for the last three annotation batches of the corpus (see Figure 3.6). A fourth annotator (annotator D) was trained to both annotate and adjudicate as was done with annotator C, although a slightly more hands-on approach with more detailed error feedback was used. Table 3.1 shows the IAA between C and D during their training period. The results are much higher than what was achieved by A and B in the development stage, but they are also lower than their results after the guidelines were completed (see Figure 3.4).

Table 3.2 presents the pairwise IAA results of all annotators on a small dataset (60 records) which was the only part of the data annotated by four of the annotators. The data is the completed part of the last batch A worked on (also completed by B), which is also the first batch C annotated. It was also purposefully chosen as the data for the final training of D. None of the annotators had seen the data before they annotated it. While still the lowest score, the agreement between C and D has significantly improved after training, and in fact they score better when paired with the other annotators. The complete agreement between A and C may seem odd, but it can be explained by the fact

that there are only 15 semantic entities in the dataset. Such a low density is not unusual for the whole corpus, as is shown in Section 3.3.3.

	Chunks						Entities					
	AB	**CD**	**AC**	**BC**	**AD**	**BD**	**AB**	**CD**	**AC**	**BC**	**AD**	**BD**
Pr_S	0.86	0.82	0.90	0.81	0.86	0.85	0.79	0.60	1	0.79	0.60	0.50
Re_S	0.84	0.75	0.91	0.84	0.78	0.78	0.73	0.90	1	0.73	0.90	0.70
$\boldsymbol{F_{1\,S}}$	**85**	**78**	**90**	**82**	**82**	**82**	**76**	**72**	**100**	**76**	**72**	**58**
Pr_R	0.90	0.92	0.90	0.84	0.94	0.92	0.79	0.67	1	0.79	0.67	0.50
Re_R	0.88	0.84	0.92	0.87	0.85	0.84	0.73	1	1	0.73	1	0.70
$\boldsymbol{F_{1\,R}}$	**89**	**88**	**91**	**86**	**90**	**88**	**76**	**80**	**100**	**76**	**80**	**58**

Table 3.2: Pairwise IAA between all annotators. The S and R subscripts stand for *strict* and *relaxed* agreement. Columns represent annotator pairs denoted by their letters.

3.3 THE HARVEY CORPUS

The Harvey Corpus is a collection of linguistically annotated de-identified clinical text. The data consists of 750 primary care patient examination notes (around 17,656 words, 23,969 tokens) with three layers of linguistic annotation. The first layer contains part-of-speech tags automatically assigned by cTAKES (Savova et al., 2010). The second and the third layers consist of manually annotated syntactic chunks and semantic entities. The rest of this section provides a description of the data selection process (Section 3.3.1), a more detailed explanation of the text processing and data manipulation that produced a single coherent data structure (Section 3.3.2), and an analysis of the annotation statistics (Section 3.3.3).

3.3.1 *Data Selection*

The Harvey Corpus data was randomly sampled from two datasets of GPRD data pooled together. The datasets were compiled for previous PREP studies, which focused on patients diagnosed with ovarian cancer (Koeling et al., 2011a,b; Carroll et al., 2012) and rheumatoid arthritis (Nicholson et al., 2013; Ford et al., 2013, 2015). The data included the records

of 344 ovarian cancer patients and 6,387 patients with rheumatoid arthritis diagnosed between 1/6/2002 and 31/5/2007, and between 1/1/2005 and 31/12/2008 respectively. The data included the records of each of the selected patients for one year before the diagnosis and two weeks after.

These samples were compiled by selecting a number of patients with the relevant diagnosis and retrieving all their records for the preceding year. Therefore, even though the Harvey source data has some diversity, it is not representative of the entire GPRD. Additionally, before the random selection, the data was filtered to remove all notes under five tokens, notes containing only test results or image attachments, and communication with specialists. The latter records were excluded because the language of letters is quite formal and detailed, which makes it completely different from the language of GP-written notes.

3.3.2 *Data Assembly*

The Harvey Corpus consists of a set of records, each about a patient encounter. Each record consists of a Read code term, followed by a sequence of tokens. The records were tokenised in two stages – before and after the annotation phrase. The first stage used simple, conservative rules to tokenise regular use of punctuation, while the second stage involved tokenisation rules that were more specific to the patterns in the text. The second stage also integrated information from the manual annotation layers to identify additional token borders. The final version of the corpus contained 750 records, 23,969 tokens, and 11,290 annotations.

The POS annotation layer was generated using the cTAKES system (Savova et al., 2010). This system was chosen because clinical text was used to train its models. The choice was further supported by the observation that the model correctly tags some idiosyncratic medical abbreviations such as *c/o* (complains of). Finally, syntactic chunks and semantic entities were manually annotated as described in Section 3.2.

3.3.3 *Data Analysis*

Compared to well-known clinical and biomedical corpora, the Harvey Corpus is quite small (see Table D.3), but it is comparable in size to the corpora of clinical text that have been linguistically annotated (Pakhomov et al., 2004; Fan et al., 2011, 2013). Table 3.4 shows annotation counts and tokens per annotation. On average, semantic entities are longer than chunks, which is to be expected from their definitions. QEs normally contain a quantity and a unit of measurement; TEs are very variable, ranging from very short jargon expressions such as *2/7* (meaning two days), to full adjunct constructions like *a month before cancer diagnosis*; and OEs are dominated by the three character abbreviation *O/E*. Only LEs tend towards a single token average, because they typically occur as modifiers to a head noun in compound nouns such as *abdomen pain*, or abbreviated in one token – ULQ (upper left quadrant). Syntactic chunks tend to be short and frequent, as a consequence of the telegraphic nature of the notes. The average number of tokens per chunk is below 1.5, which is indicative of a very large proportion of single token chunk annotations. While this is to be expected from MVs and APs, the frequency and brevity of NPs certainly reflects the qualities of this kind of clinical language.

Chunk	**NP**	**MV**	**AP**	**CHs**	**TE**	**LE**	**QE**	**OE**	**SEs**	**All**
Pr_S	0.87	0.89	0.68	0.85	0.77	0.65	0.84	0.95	0.74	**0.83**
Re_S	0.87	0.89	0.78	0.87	0.70	0.63	0.65	0.96	0.68	**0.84**
$F_{1\ S}$	87	89	73	86	74	64	73	95	71	**84**
Pr_R	0.93	0.90	0.73	0.90	0.91	0.74	0.90	0.96	0.84	**0.89**
Re_R	0.93	0.90	0.84	0.92	0.83	0.72	0.69	0.97	0.77	**0.89**
$F_{1\ R}$	93	90	78	91	87	73	78	97	80	**89**

Table 3.3: Harvey Corpus Statistics: strict and relaxed inter-annotator agreement measured as f-score. The performance for the two groups, chunks (CHs) and semantic entities (SEs), was measured using micro-averaging.

Another aspect of the data that highlights the gap between the frequency of NPs and the other annotation types is the number of records with more than five occurrences of a single annotation type. The figures in Table 3.4 suggest that only NPs and MVs are likely

to occur more than 5 times in a single record. This is confirmed in Figure 3.5 which shows the number of records with the number of each type of annotation.

The inter-annotator agreement shows a continuation of the positive trend from the training stage across the nine batches into which the corpus was divided for the annotation process (see Figure 3.6). The relatively large difference between the strict and relaxed agreement scores for most annotation types (5 percentage points on average, see Table 3.3) shows that a significant amount of the conflicting annotation could be overcome with minimal intervention during the adjudication process. This provides further evidence of the good quality of the final corpus annotation. The agreement improvement varies from less than 1% (OEs) to over 13% (TEs) depending on the characteristics of the annotation types. Main verbs are much less prone to chunk boundary disagreement, because in most cases they are a single word. On the other hand, the boundaries of temporal expressions could be difficult to identify with confidence in more complex cases such as periods of time (e.g. *more than six months*).

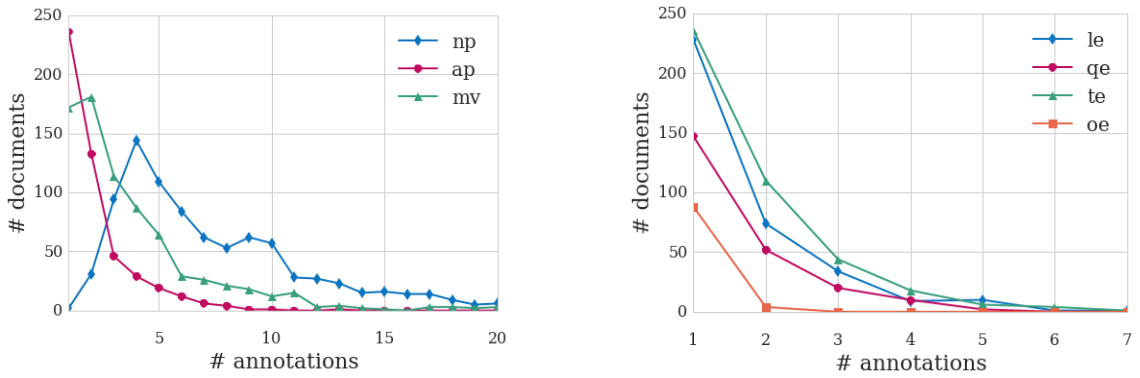


Figure 3.5: Distributions of annotations by annotation type: chunks (a) and semantic entities (b).

Chunk	**NP**	**MV**	**AP**	**CHs**	**TE**	**LE**	**QE**	**OE**	**SEs**	**All**
Count	6,304	2,613	893	9,810	605	481	321	73	1,480	11,290
Tok/Ann	1.61	1.00	1.18	1.41	1.66	1.34	1.49	1.13	1.49	1.52
Ann/Rec	8.40	3.48	1.19	13.08	0.81	0.64	0.43	0.10	1.97	15.05

Table 3.4: Harvey Corpus Statistics: annotation counts, average tokens per annotation, and average annotations per record

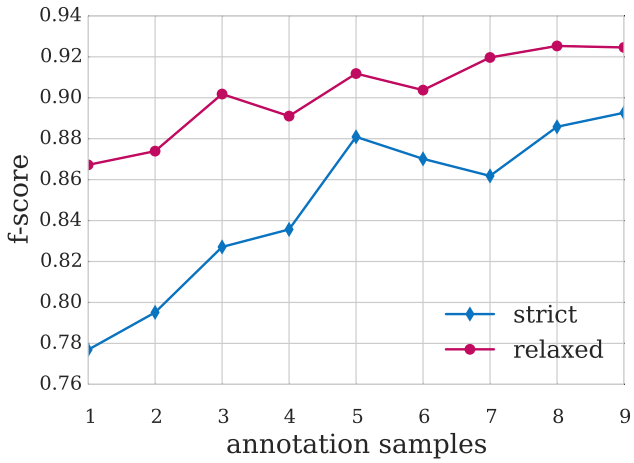


Figure 3.6: Inter-annotator agreement for the nine annotation batches of the corpus, in the order they were annotated.

3.3.4 *Corpus Availability*

The data that the Harvey Corpus was drawn from was obtained as part of the Patient Records Enhancement Programme under a licence from the GPRD. Currently it is not permitted to share any of the data with anyone not covered by this licence agreement. However, the PREP team is working towards public release of the data. Meanwhile the annotation guidelines as well as the annotation (without the text) are available for download on GitHub[3].

3.3.5 *Additional Data*

The annotator training process generated a considerable amount of annotated data. Most of those annotations cannot be considered of good quality. However, the ones created in the later stages of training are comparable with the quality of the corpus, except for the way they were adjudicated. But since they were of generally good quality, it was decided that they should be included in the corpus.

The first of these datasets was created during an annotation workshop organised for the first three annotators. Each of them annotated thirty GP notes, a third of the dataset (ninety notes, 728 tokens), and then passed them on for review to the next annotator. In this way, the data was annotated once and reviewed twice. During the review the annota-

3 https://github.com/savkov/harvey-corpus

tors were allowed to change the annotation in any way, and were given the opportunity to discuss changes with their colleagues.

The second dataset included in the corpus was generated in the final stages of the training of the last annotator, later used as an adjudicator. Fifty GP notes (2,073 tokens) were annotated by the medical student, and then the annotation was discussed and improved in cooperation with a computational linguistics expert (the author of this thesis).

Chunk	**NP**	**MV**	**AP**	**CHs**	**TE**	**LE**	**QE**	**OE**	**SEs**	**All**
Count	7,234	2,915	1,034	11,183	741	612	369	97	1819	13,002
Tok/Ann	1.64	1.02	1.20	1.44	1.68	1.41	1.51	1.13	1.53	1.45
Ann/Rec	8.13	3.28	1.16	12.57	0.83	0.69	0.41	0.11	2.04	14.61

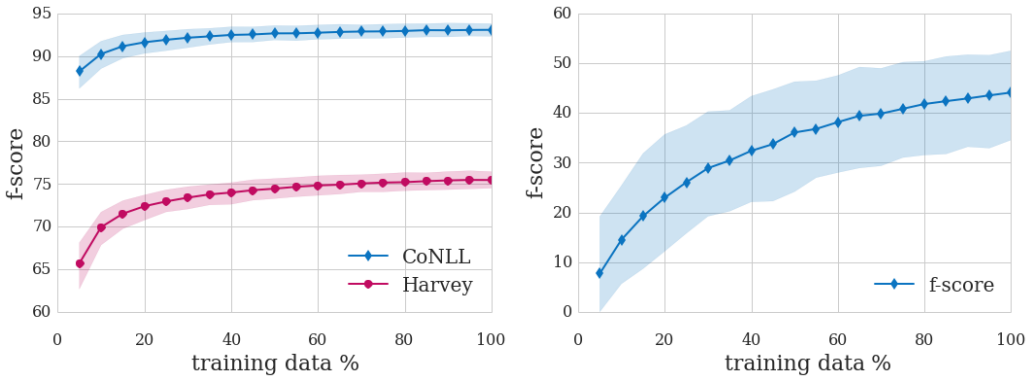
Table 3.5: Extended Harvey Corpus Statistics: annotation counts, average tokens per annotation, and average annotations per record

Table 3.5 shows a version of Table 3.4 updated with the additional data described above.

3.4 EXTRINSIC EVALUATION

The lack of an established quality metric for annotated corpora makes it difficult to compare and evaluate them. Therefore, corpora are often extrinsically evaluated through the impact they make on an application task. Following this methodology, experiments were set up to evaluate the performance of two statistical models trained on Harvey Corpus data: one for chunking, and one for entity recognition. A comparison experiment was also set up using a randomly selected dataset (of size comparable to the Harvey Corpus) extracted from the Penn Treebank chunk data from CoNLL-2000. YamCha (Kudo and Matsumoto, 2001, 2003), a widely-used SVM-based sequential tagger, was used to generate the models in all three experiments. The first two experiments aimed to establish if the corpus provides enough training data to achieve adequate results for the tasks of syntactic chunking and entity recognition. The third experiment aimed to compare the learning rates and the difference in performance between the Harvey data chunking model

and one trained on edited text. The experiments with Harvey data were based on a 90:10 split of the 750 annotated records into a training and validation set.



(a) chunking (b) semantic entity recognition

Figure 3.7: 500-fold bootstrapping learning curves generated using YamCha. Shaded regions indicate 95% confidence intervals. The subsampling process used for the bootstrapping was always sampled from the same training set, and the evaluation was always performed on the same validation set using the evaluation script from CoNLL-2000.

Figure 3.7 shows the accuracy of the models estimated using bootstrapping (Efron, 1979, 1983; Efron and Tibshirani, 1997) as the training data size increases. Instead of repeatedly analysing disjoint subsets of the data, as in cross-validation, bootstrapping repeatedly analyses sub-samples. Each sub-sample is a random sample with replacement from the full sample. The number of used sub-samples typically should reach the number of data points but in some cases that is not necessary depending on the task. Each data point on the curves represents the mean f-score of five hundred repeated evaluations using bootstrapping. As a result, the average standard error of the mean is low: 0.14 base points for the chunks curve, and 0.30 for the semantic entities curve.

The monotonically increasing learning curve of the Harvey Corpus chunking, and the decreasing standard deviation suggest that the corpus contains consistent chunking annotation, supporting a stable learning process. The increasing curve trend indicates that more training data should improve the performance, but it is difficult to predict to what extent. The gradient of the Harvey Corpus learning curve is very similar to that of a model trained on the Wall Street Journal, but the absolute performance is much lower. The experiment did not try to adjust the training process in any way, but used the standard YamCha feature set (Kudo and Matsumoto, 2001) and evaluation. Improving the quality of the POS tags of the Harvey Corpus and tuning the features may provide performance improvement.

Note that the model used automatically generated POS tags using the cTAKES model (Savova et al., 2010). It is also likely that the proportion of unknown tokens encountered by the clinical text model is much higher than that of the Penn Treebank model, which leaves more room for improvement through techniques tackling that issue.

On the other hand, the entity recognition model has a steeper learning curve, but a much lower final performance of 0.43 f-score. However, these results are promising, because the distribution of entity annotation is less balanced and much less frequent than that of the syntactic chunks, which is more uniform and covers about 60% of all tokens (see Table 3.4 and Figure 3.5). A closer look at the results shows that locative expressions are recognised with low accuracy, achieving only 25% correctly tagged tokens, as opposed to over 90% for on-examination expressions and 55% for temporal and quantitative expressions. This can be explained by the very large vocabulary of the locative expressions, including body parts and regions expressed in both conversational and medical language style.

3.5 CHAPTER SUMMARY

This chapter described the development of a set of annotation guidelines and an annotated corpus of primary care clinical records consisting of physician-typed free-text notes and Read codes. It discussed the background, motivation, and data source of the corpus as well as an evaluation of its annotation quality.

Since the chunk annotations of most established language resources have been automatically generated rather than hand-annotated, the chunk annotation guidelines presented in this study are without parallel for the English language. They were planned as a self-sufficient tuition instrument specifically for use by domain experts. They contained enough easily digestible linguistic knowledge to support the annotation process. Their development and the annotator training were set up as iterative processes, with annotation accuracy improving on each iteration. It was found that experience and longer annotation sessions improves IAA, while long periods of time between annotation sessions result in deterioration. After the training process was complete, inter-annotator agreement reached 86 f-score for annotation of chunks, 71 for semantic entities, and 84 overall. The resulting parallel annotations of the corpus were combined by a third domain expert resolving the conflicts with minimal intervention, producing the final version of the Harvey Corpus, con-

taining 750 records, 23,969 tokens, and 11,290 annotations. The corpus was extrinsically evaluated using two practical machine learning tasks, showing that its chunk annotation is consistent and reliable (although the semantic entity annotation is not sufficient for training an accurate classifier). The experiments showed that performance increases with more training data and that the learning rate of the chunking classifier is comparable (but with a lower starting point) to a classifier using data from the CoNLL-2000 data set.

Despite these positive results, there are limitations to the Harvey Corpus: its relatively small size compared to other clinical text corpora, and lack of other important annotation layers such as parts of speech. Even though adding more data seems unlikely to increase chunking accuracy to levels seen with edited text, it is evident from the learning curves that adding more data will improve accuracy. Addressing other issues, such as POS tagging errors, should also decrease the chunking error rate, as its imperfect quality could have a harmful effect on the decisions made by the classifier. However, quantifying that effect requires a much more detailed analysis of the relation between the two. Such analysis should also optimise the features of the models for primary care data, as the configuration used in this study was the optimal YamCha configuration for the CoNLL-2000 data.

While the Harvey Corpus is the first annotated language resource based on UK primary care text large enough to be used for developing machine learning tools, there are previous studies on US secondary care data with comparable goals. Both this study and that of Fan et al. (2013) are essentially aiming to add syntactic information to difficult to process clinical text, but using different approaches and different data. It is difficult to compare results as there is free access only to the annotation, but not the textual data of their study. However, the learning curve suggests that if more data is available the chunking accuracy may reach 80, which is comparable to the performance of Fan et al.'s constituency parser. Even so, a fair evaluation would require an extrinsic measurement, such as impact on symptom identification, since chunking and constituency parsing are evaluated in very different ways.

In conclusion, the Harvey Corpus provides a shallow parsing gold standard for physician-typed clinical notes text, which allows the development of accurate tools for syntactic chunking. The accompanying guidelines are a unique resource that allows annotation of clinical data to be carried out for future research. The corpus and annotation guidelines

can support future research in processing this kind of text and could serve as a foundation layer for annotating medication, symptoms, and diseases.

4

CORE NATURAL LANGUAGE PROCESSING

This chapter discusses the application of part-of-speech tagging and chunking to primary care text. The output of these processing steps is a key factor in developing a reliable processing pipeline, because they are the basis for the most important features used in higher level tasks such as named entity recognition (NER), and in the case of this thesis — medical concept recognition.

The experiments described in this chapter follow the previously proposed strategy concentrating on developing a machine learning model for chunking of primary care text. The necessary resources for the full development of a POS model were not available, so a number of widely used part-of-speech models representing a range of different approaches, were included in the development process.

The chunking research and associated model presented in this chapter aimed to answer four questions: 1. how successful are standard models applied without any customisation to the task (Section 4.1); 2. how well would the available tools and techniques deal with the challenges of the task at hand if they were optimised for it (Section 4.2); 3. will bespoke new features improve performance substantially, and which ones exactly, e.g. what data should be used for word representation features, what type of suffix features work best for primary care text (Section 4.3); 4. what is the most suitable feature set for the task given the process may involve different classifiers and part-of-speech annotation (Section 4.4).

4.1 APPLYING EXISTING TECHNOLOGY

A natural first step in developing specialised statistical NLP models is evaluating and analysing the performance of freely available general domain tools and models applicable

to the task. This is necessary as it should determine the need for a new specialised model, and the issues with existing technology that should be addressed by such a model. This section describes the first experiments with Harvey Corpus data, and the limitations of using models and tools trained on general domain data, or datasets with some similarities to UK GP notes.

4.1.1 *Approximate Evaluation of Part-of-Speech Processing*

The idea of annotating the corpus directly with higher level annotation layers without part-of-speech tags relied on the observation-based conviction that the current state-of-the art tools are capable of providing an adequate POS processing of the data. The POS tagger performance could be judged based on two different types of outcomes: the traditional evaluation against a gold standard, which would require further annotation; or measured extrinsically as the improvement of the performance of a higher level (chunking or NER) model evaluated against its respective gold standard. While the former method is much more reliable and informative than the latter, as discussed in Chapter 3, it requires a significant effort into creating an evaluation set of good quality. Perhaps the best way to approach part-of-speech evaluation in this situation is to use extrinsic methods as the main measurement. At the same time, it is a good idea to produce a small POS annotated subset of the Harvey Corpus in order to get a sense of the absolute performance rate, and to a certain extent validate the performance assumptions.

The author of this thesis annotated 100 records (2,077 tokens) following the Penn Treebank annotation guidelines (Santorini, 1990). A range of freely available part-of-speech tagging models were selected for an evaluation experiment on that dataset. The selection was made so that it represents a wide range of approaches to the task, while using established tools and models. The evaluation process over the small dataset showed that the models achieved f-score ranging between the 70s and the lower 80s (see Table 4.1). It seemed only natural that the best performing taggers are cTAKES (Savova et al., 2010), which is the only one with training set containing clinical text, GENIA (Kulick et al., 2004a; Tsuruoka et al., 2005; Tsuruoka and Tsujii, 2005), which is the state of the art in processing biomedical literature, and the POS tagger from the Stanford NLP toolkit (Toutanova et al., 2003).

Model	Accuracy
ARK_{NPS}	75.43%
ARK_{Ritter}	75.72%
cTAKES	**82.40%**
GENIA	80.61%
Stanford	80.69%
SVMTool	76.41%
Wapiti	73.37%

Table 4.1: Accuracy of POS tagging models evaluated on 100 manually annotated records from the Harvey Corpus.

The ARK tagger (Gimpel et al., 2011; Owoputi et al., 2013) models (one trained on the NPS Chatroom Conversations corpus, and one trained on the corpus presented by Ritter et al. 2010[1]) were also included in the experiment in the expectation that the robustness needed for processing tweets could be of use when processing Harvey Corpus text. It is interesting that their results are comparable to those of the SVMTool (Giménez and Màrquez, 2004) and Wapiti (Lavergne et al., 2010) models. However, the difference in accuracy between the Stanford model and the other models trained on the Penn Treebank (Wapiti and SVMTool) seems too large given that they were trained on the same data with slightly differing feature sets, and the fact that the structured classifier (Wapiti) does not have the advantage. Only the Stanford model uses word representation features (distributional similarity word clustering based on "POS induction" Clark, 2013), but even so it seems unlikely that it should be the sole reason given that the thesaurus, from which the features were extracted, was also generated from the Penn Treebank.

The models can be clustered into two groups based on their accuracy. The difference between the groups is quite large, but some of those within the groups were quite small. There are a number of ways to test if such differences are statistically significant. Computationally-intensive randomisation tests are a type of stratified shuffling (Noreen, 1989, Chapter 2) similar to the matched-pairs t-test (Cohen, 1995, Section 5.3.2). Under the null hypothesis of this method the two compared techniques should be the same, so any output that comes from one of them could just as well have come from the other. Therefore, if the paired outputs are shuffled, very little should change in the overall evaluation metric. If the process is repeated for all possible ways that can be done, the results can be used to

1 Throughout this thesis these corpora are referred to as *the NPS corpus* and *the Ritter corpus*.

estimate the probability of a difference at least as large as the initial difference between the methods occurring by chance. If that probability is found to be too small, then the null hypothesis is rejected. Yeh (2000) suggest that such a way of computing the test is tractable for items $n < 20$, as the total number of iterations should be 2^n (1,048,576). For the case of $n > 20$ an approximation was suggested where each shuffle is performed with random assignments. That is all output items of the compared methods are iterated through, tossing a coin to decide if they should be swapped, and repeating that 2^{20} times.

	cTAKES	**GENIA**	**NPC**	**Ritter**	**Stanford**	**SVMTool**
GENIA	P<0.092	-				
NPC	P<0.001	P<0.001	-			
Ritter	P<0.001	P<0.001	P<0.470	-		
Stanford	P<0.089	P<0.791	P<0.002	P<0.001	-	
SVMTool	P<0.001	P<0.001	P<0.972	P<0.609	P<0.001	-
Wapiti	P<0.001	P<0.001	P<0.079	P<0.187	P<0.001	P<0.023

Table 4.2: Statistical significance between pairs of POS models evaluated on Harvey Corpus data. *P*-values were calculated using approximate randomisation test.

The *p*-values in Table 4.2 were calculated using approximate randomisation tests with one million repetitions. The table shows that only model pairs from different clusters have achieved significant difference in their accuracy.

The accuracy of the models presented in this section confirms that part-of-speech taggers perform at a level much lower than their general domain achievements (e.g. on the Wall Street Journal part of the Penn Treebank). However, whether that level is acceptable for developing a chunking model, which is the main focus of this chapter, is difficult to determine without experimentation.

4.1.2 *Approximate Evaluation of Existing Chunking Models*

A fully reliable evaluation of the chunking models trained on other resources is unfortunately impossible due to the annotation scheme used in the Harvey Corpus. The CoNLL-2000 scheme includes prepositional phrase chunks, and the noun phrase chunks and verb chunks are defined differently. However, since the definition of NP chunks used in Harvey

is base NPs, which is a subset of the NP chunks in the CoNLL-2000 dataset, it was worth testing the accuracy of established models trained on that dataset at least in that isolated case. Even though the tests would not be fair from multiple perspectives (different number of tags in each tagset; very different distribution of the `outside` tag), it was important to investigate whether existing chunking models could be used as a realistic baseline for models optimised specifically for the Harvey Corpus data. In this experiment, existing chunking models were used to annotate the Harvey Corpus, and their output was compared to the gold standard annotation. All models used POS tag features, which were obtained using the cTAKES POS tagging model already used for the experiments in Section 3.4.

The evaluation was conducted by transforming all chunk annotations other than NPs into OUTSIDE tags in both the test and gold standard outputs, and then calculating precision and recall using the CoNLL-2000 evaluation script. The left results column in Table 4.3 shows the f-score of four chunking tools, ranging from low to the mid 40s. The GENIA and cTAKES taggers were selected because of their training data, and the YamCha (Kudo and Matsumoto, 2003, 2001) and CRF++[2] as a representation of structured and binary prediction algorithms. GENIA shows the best performance by nearly 2 percentage points, followed by YamCha, and CRF++, which have been developed by the same author. Surprisingly the performance of cTAKES (Savova et al., 2010) in this case is much lower than GENIA.

	Harvey		**WSJ**
	All	**NPs**	**All**
GENIA	46.41	n.a.	n.a.
YamCha	44.39	47.84	93.91
CRF++	43.76	46.32	93.83
cTAKES	42.57	n.a.	n.a.

Table 4.3: F-scores of the four chunking models on the Harvey Corpus, using the full tagset, or only the NP annotations, compared to their accuracy measured on the CoNLL-2000 test set.

A natural improvement of this experiment was to also isolate the NPs in the training data, i.e. convert all other annotations to OUTSIDE annotations. Again, an obvious flaw

2 CRF++ is an established CRF-based tool by Taku Kudo. It can be downloaded under the LGPL license from http://taku910.github.io/crfpp/

of this approach is the "annotation" of many noun phrase chunks as non-chunks, because they were part of a prepositional phrase chunk. Additionally, this experiment could not be performed with all available tools for technical reasons: to the best of this author's knowledge the publicly available software of the GENIA tagger does not support training a new model, while the cTAKES tagger was trained on the MED corpus, which was not available for this study. A significant improvement in the performance of YamCha and CRF++ were achieved when their training data was manipulated (see Table 4.3).

It is difficult to determine what part of the errors made by the models were caused by the mismatch of definitions of NPs, and other plausible reasons, such as unknown or missing words, and ungrammatical constructions. However, the accuracy of the models is only around half of what they achieve on text similar to their training data, and about 30 base points less than those of the models demonstrated during the extrinsic evaluation of the Harvey Corpus in Chapter 3. Such a stark difference between models trained on general and clinical text highlights the need for in-domain training data when processing primary care text such as that in the Harvey Corpus.

4.1.3 *Training Models with Standard Tool Configurations*

In the previous two sections it was established that using readily available chunking models has very low accuracy compared to an in-domain model using a training configuration (see Chapter 3), while the selected group of POS tagging models achieved f-scores much lower than their accuracy on text similar to their training data. This section describes an experiment comparing models trained on the Harvey Corpus using YamCha and CRF++ with their respective best-performance model training configurations. The part-of-speech annotation plays an important role in forming the feature vectors of machine learning models, so the experiments were replicated for each of the POS tagging models listed in Section 4.1.1, and two other models using different tagsets — the POS tagger part of the RASP parser (Briscoe et al., 2006) and an ARK model using a specific tagset tailored for tweets (Gimpel et al., 2011). Each model was evaluated using ten-fold cross-validation on the whole Harvey Corpus.

The results in Table 4.4 show that the effect of POS annotation on chunking performance is different from what would be expected from the results discussed in Section 4.1.2. The

	YamCha			CRF++		
	Pr	**Re**	**f-score**	**Pr**	**Re**	**f-score**
ARK_{NPS}	75.51	74.34	74.89[a]	76.02	75.16	75.57[c,d]
$ARK_{Twitter}$	**76.22**	**76.23**	**76.23**	**76.74**	**76.15**	**76.51**[e]
ARK_{Ritter}	76.53	74.91	75.68	**77.00**	**75.61**	**76.32**[e]
cTAKES	75.37	73.66	74.59[b]	76.11	74.54	75.28
GENIA	73.59	71.54	72.52	74.39	72.76	73.61[f]
RASP	75.43	73.71	74.62[b]	76.45	74.84	75.62[d,g]
Stanford	75.60	74.11	74.79[a]	76.28	75.27	75.71[c,g]
SVMTool	74.45	72.76	73.61	75.15	73.58	74.33
Wapiti	74.15	71.57	72.91	74.58	72.62	73.66[f]
baseline	67.14	60.21	63.53	69.98	65.09	67.34

Table 4.4: Impact of part-of-speech annotation on chunking using various POS models and chunking configurations. The *baseline* is trained without POS annotation features. The result values are the mean outcomes of ten repeated 10-fold cross-validation experiments. Values with matching superscripts do not differ significantly from one another.

performances of the ARK tagger models are much better than the expectations based on the previous evaluation, while the GENIA results shift from one of the highest to one of the lowest. The fact that a model using a simple tweet-bespoke tagset delivers the top performance on primary care data, suggests that re-training some of the available POS models with a simple POS tagset may lead to improvements in chunking accuracy.

As the majority of the results are quite close, it is not obvious which differences are statistically significant. As it cannot be assumed that the population variance of each sample (set of repeated experiments) is the same, Welch's t-test (Welch, 1947) was performed on each pair of result averages. The null hypothesis for each test was that the averages are not significantly different from each other. Using a conservative 0.01 p-value, that hypothesis was rejected for most of the tested pairs except for ones marked with matching superscripts in Table 4.4. The tests were also performed using the more relaxed 0.1 p-value finding that two more pairs rejected the null hypothesis. These tests confirmed that the YamCha model trained with $\text{ARK}_{\text{Twitter}}$ POS annotation performs significantly better than all other YamCha models, which supports the suggestion that a simpler POS tagset might improve chunking. However, in the case of the CRF++ model, the advantage may not be significant, since there is little difference in accuracy between the model using tags produced by $\text{ARK}_{\text{Twitter}}$, and by $\text{ARK}_{\text{Ritter}}$.

Section 4.1.2 showed little difference in the accuracies of the YamCha and CRF++ chunking models. The series of experiments in this section show that CRF++ consistently achieves slightly higher results than YamCha. The differences were found significant using Welch's t-test on all model pairs using the same POS annotation. This proves that using the feature sets suitable for the news domain (Penn Treebank), CRF++ performs better than YamCha. However, at this stage it is not clear if these feature sets are optimal for the data at hand.

4.2 OPTIMISING AVAILABLE CHUNKING MODELS

The evaluation reported in the previous section showed that by simply re-applying the training process of successful chunking models, we could achieve reasonable accuracy on the Harvey Corpus. However, these results could be potentially improved through optimising the machine learning features and hyperparameters, as well as exploring other options. For example, the chunk representation scheme (the way chunks are broken down into token level annotations) could have an influence on the results, as well as the type of machine learning algorithm used in the chunking tools, or the part-of-speech tagset of the training data.

Most approaches to the analysis of natural language base their processing on the tokens (words) and other levels of linguistic annotation from the context in which each token occurs. Normally the final stage of optimising a machine learning system for a particular problem (or kind of data) involves the fine tuning of these context features and the algorithm hyperparameters. The tuning process, also referred to as (model) development stage, may test certain ideas logically motivated by given features of the data, or conclusions drawn from an error analysis, but others, e.g. hyperparameters, may simply be optimised through a trial and error approach.

Considering the potential improvement avenues, a series of experiments aiming at three potential sources of improvement were conducted. All experiments sought the optimal configuration of context features and hyperparameters within a reasonably limited search space. Firstly, as shown above, POS models may have a significant impact on chunking performance, so the experiments explored a wide range of POS annotation. Nine POS taggers were used covering various machine learning techniques and types of training data

Name	Algorithm	Training Data
ARK	CRF	ARK tweet dataset (Owoputi et al., 2013)
ARK_{NPS}	CRF	NPSChat IRC (Forsythand and Martell, 2007)
ARK_{Ritter}	CRF	Ritter tweets (Ritter et al., 2010)
cTAKES	MaxEnt	Mayo Clinic (Pakhomov et al., 2004), GENIA, PTB
GENIA	MEMM	WSJ, GENIA, PennBioIE
SVMTool	SVM	WSJ
Stanford NLP	MaxEnt	WSJ, PennBioIE, small custom datasets
Wapiti	CRF	WSJ

Table 4.5: List of POS tagging models used in the chunking feature optimisation experiments.

(see Table 4.5). Secondly, the differences in performance between SVM-based and CRF-based chunking models were also explored. Finally, the implications of choosing between two chunk representation patterns were investigated.

4.2.1 *Experimental Setup*

Before conducting experiments, it was important to settle on an evaluation scheme to be used across the whole model development, ensuring that only the final model was tested on the unseen validation set. Inner cross-validation (Azzalini and Scarpa, 2012) is an evaluation scheme (see Section 2.3.4) that was used for all model development experiments. The Harvey Corpus was split into a development set, which was approximately 90% of the data, and a test set made up of the remaining 10%. Each development experiment uses cross-validation (with a 90:10 split of the development set) for evaluating different features, while the final evaluation process (see end of Section 3.4) uses the whole development set for training and the test set for evaluation.

The main method of optimising the performance of a model feature set in most experiments is searching across all context window sizes and other parameter values of the feature types (e.g. POS annotation). Due to the high number of feature types, an unrestricted search space would have been intractable, so a global restriction on the width of the context window was imposed during experimentation. There was little preliminary ev-

idence that a window wider than three tokens in each direction would be helpful, agreeing with the observation that there is little global structure within each text record.

Hyperparameters were generally not tuned during most of the development experiments to avoid combinatorial explosion. The intuition was that the process of searching for the optimal feature vector should precede the tuning process, so the latter is only done at the end of the development process.

The use of cross-validation during development made the task of testing the significance of result differences less complicated, because it involved ten repetitions of each evaluation on different data splits. Then the comparison of the two experiments could be regarded as a comparison between the means of two groups of repeated measurements. Since the sample pairs contain only ten observations and their distribution is unknown, the Wilcoxon signed-rank test, referred from here on simply as the Wilcoxon test (Wilcoxon, 1945), is the most appropriate choice. The p-value threshold that indicates statistical significance for the development experiments is set to $\leqslant 0.01$.

4.2.2 *CRF++ vs. YamCha*

Even though the various machine learning algorithms have advantages in different circumstances and problem settings, it is difficult to single out only one algorithm that will always deliver top performance. This section compares the two most widely used machine learning tools for sequential tagging problems: CRF++ and YamCha. Normally, a structured algorithm such as conditional random fields is expected to have an advantage in solving sequential tagging problems due to its global optimisation, but each record has little global structure, which may potentially negate the aforementioned advantage.

To determine the better option, a new feature tuning experiment was set up using both tools optimising over all supported feature types within a context window of three in each direction. The results in Table 4.6 confirm that the CRF-based tool achieves better performance with all considered POS annotations and against the baseline, although not all differences are statistically significant if a strict 0.01 p-value is assumed.

	Precision		Recall		F-score		
POS	C	YC	C	YC	C	YC	*p*
ARK_{NPS}	77.65	74.84	76.87	73.71	77.26	74.26	0.007
ARK_{Ritter}	76.23	73.51	74.19	70.93	75.20	72.20	0.007
$ARK_{Twitter}$	78.15	75.28	77.40	75.52	77.76	75.40	0.047
cTAKES	77.47	75.00	75.40	73.16	76.42	74.06	0.059
GENIA	76.62	72.77	74.79	70.82	75.68	71.77	0.007
RASP	77.58	74.59	75.83	72.88	76.69	73.72	0.007
Stanford	77.90	75.01	76.59	73.67	77.23	74.33	0.028
SVMTool	76.73	73.83	75.22	72.31	75.97	73.06	0.012
Wapiti	76.04	73.36	74.26	70.70	75.14	72.00	0.022
baseline	71.39	68.06	66.74	62.20	68.98	64.99	0.009

Table 4.6: Comparison between CRF++ (C) and YamCha (YC) chunkers trained on the Harvey Corpus using different POS annotations. The p-values were calculated using the Wilcoxon test.

4.2.3 *Chunk Representation*

A chunk is by definition a multi-token entity, so it can be represented with a single label or entry only using stand-off annotation that refers to its scope with text position indices. Such a style of annotation is also suitable for representing relations, and it is usually chosen for corpora with a number of different annotation layers. However, a more suitable representation for natural language analysis tools that functions on the token level breaks chunk annotation down into a pattern of token labels, rather than a single chunk label.

The chunk representation pattern BIO, which is most commonly used, stands for *beginning, inside, outside*. It takes a minimalistic approach to the representation problem in order to keep the number of labels low. It was introduced by Ramshaw and Marcus (1995) and later established in the NLP community with the CoNLL-2000 shared task (Tjong Kim Sang and Buchholz, 2000). Note that for chunking representations the total number of labels is the product of the chunk types and the set of representation types plus the outside tag, meaning that for BIO with a set of three chunk types such as the one used for the Harvey Corpus (NP, MV, AP) there will be seven labels: B-NP, I-NP, B-AP, I-AP, B-MV, I-MV, and O.

Alternatively, chunks could be represented using a more fine grained pattern such as BEISO[3], standing for *begin, end, inside, single, outside*, as was used by Kudo and Matsumoto (2001). Such a pattern naturally increases the cost of learning due to the greater size of the tagset, and it could be considered unnecessary, because it is deterministically inter-convertible with the BIO representation pattern. However, it could be useful in cases where a more fine-grained tagset could pick up more detailed disambiguation information. For instance, an *end* tag could be useful for better recognition of boundaries between consecutive chunks of the same type. In this case, the BEISO tagset model would consider the boundary before and after crossing it, while a BIO model would only consider it after. This difference should return only a small gain with standard edited text, because the chunk type distribution is more balanced, and punctuation divides ambiguous cases such as lists of compound nouns. However, the Harvey Corpus is fairly NP-heavy, and contains many sequences of NP chunks without any punctuation or other markers between them to indicate their boundaries.

	Precision		**Recall**		**F-score**		
POS	BIO	BEISO	BIO	BEISO	BIO	BEISO	p
ARK_{NPS}	77.65	77.05	76.87	78.29	77.26	77.66	0.445
ARK_{Ritter}	76.23	74.52	74.19	75.05	75.20	74.78	0.508
$ARK_{Twitter}$	78.15	77.21	77.40	78.88	77.76	78.03	0.721
cTAKES	77.47	76.86	75.40	76.82	76.42	76.84	0.678
GENIA	76.62	75.47	74.79	76.43	75.68	75.94	0.646
RASP	77.58	76.21	75.83	76.99	76.69	76.59	0.646
Stanford	77.90	77.20	76.59	78.26	77.23	77.72	0.575
SVMTool	76.64	75.57	75.13	76.60	75.97	76.08	0.959
Wapiti	76.04	74.76	74.26	75.27	75.14	75.01	0.721
baseline	71.39	70.06	66.74	69.31	68.98	69.67	0.285

Table 4.7: Comparison between chunking models using different chunk representation schemes.

The two chunk representations were evaluated in combination with each POS tagger, and the results in Table 4.7 show that for each POS model there is no significant difference in chunk f-score between BIO and BEISO representation. Although most of the achieved

3 Sometimes also abbreviated as IOBSE.

results are in favour of the BEISO representation, given that the BIO representation has been established as a standard in the field, and the lack of significant difference in performance, it was decided that further experiments should use BIO.

4.2.4 *Result Summary & Discussion*

The optimal model configuration in the experiments above uses *CRF++* with BIO chunk representation, inner cross-validation development process, and one of the four POS models: $ARK_{Twitter}$, *RASP*, *cTAKES*, or *Stanford*. The SVM-based *YamCha* was decisively outperformed by *CRF++*. Although the BEISO chunk representation configurations had higher scores in most cases, they were not significantly different from their BIO counterparts.

Even though the optimisation experiments described above aimed to be as thorough as possible, there are certain limitations that prevented the exploration of other interesting kinds of models. The chunking tools offer only a limited range of essential feature types, and they lack support for the recently proposed word representation features such as word clusters and embeddings. Therefore, in order to explore further possible improvements, experimentation with more flexible tools was required.

4.3 FURTHER FEATURE ENGINEERING

Drawbacks of the chunking tools used for the experiments in the last section are the lack of integrated semantic word representation feature capabilities, and the lack of flexibility in defining new context features. Given that the error rate of the chunking models remained at a level where one in five chunks is incorrect, it was important that all possible improvement avenues were pursued. In order to explore the influence of word representation and a number of less commonly used, yet potentially useful features, a new software tool was needed. This more flexible tool was used in a series of proof-of-concept experiments to determine whether the suggested additional features were beneficial to the performance of chunking models on primary care data.

The purpose of those experiments was not to find the best possible model using these features, but to make a quick assessment of their potential. Therefore, they concentrate on the effects of adding new feature types using the same narrow context window ([-1:0]).

The rest of this section begins with a brief description of a new feature extraction software package developed as part of this research (Section 4.3.1). Section 4.3.2 explores the effects of using a simpler tagset. Then a series of commonly used features are tested in Section 4.3.3. And finally, Section 4.3.4 discusses the use of a number of publicly available word representation clusters, as well as some newly generated from the Harvey Corpus and other available primary care text.

4.3.1 *CRFSuite Feature Extractor*

Existing chunking and POS tagging software tools integrate the extraction of features from data with the core machine learning algorithm. This integration improves the user-friendliness, but limits the kinds of information represented in the feature vectors. For the next set of experiments it was therefore necessary to identify a suitable separate core machine learning-backed software tool, and implement an interface to a feature extractor, in order to allow more freedom in how feature vectors are constructed. *python-crfsuite*[4] is a Python binding for the *CRFSuite* sequence tagger tool (Okazaki, 2007). It is one of the fastest CRF tools available (see Figure 4.1).

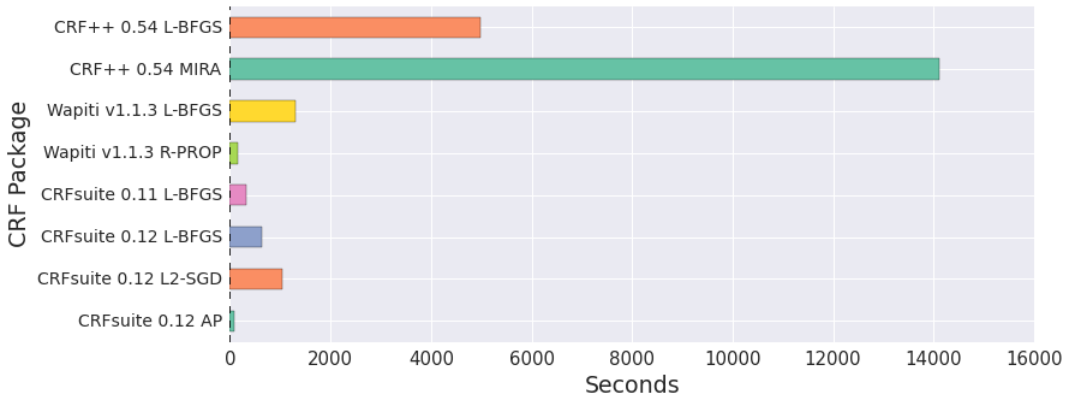


Figure 4.1: Training speed benchmark comparing *CRFSuite* to *Wapiti* and *CRF++*. Data acquired from http://www.chokkan.org/software/crfsuite/benchmark.html.

4 *python-crfsuite* homepage is at https://github.com/tpeng/python-crfsuite

A new feature extractor with various feature types was implemented to match the input of the *python-crfsuite* library[5]. The feature extraction software was designed to allow extensibility for further types of features using Python's ability to pass functions as objects.

The initially designed features included word representation clustering (plain numbers), Brown clustering (Brown et al., 1992), word embeddings (Collobert and Weston, 2008; Huang and Yates, 2009), and a number of binary features, canonicalisation features, and suffix features.

	CRF++	**CRFSuite**	p
ARK_{NPS}	77.26	77.42	0.931
ARK_{Ritter}	75.20	75.62	0.104
$ARK_{Twitter}$	77.76	77.87	0.891
cTAKES	76.42	77.05	0.049
GENIA	75.68	76.05	0.113
RASP	76.69	76.71	0.945
SVMTool	75.97	76.29	0.887
Stanford	77.23	77.44	0.829
Wapiti	75.14	75.28	0.918
baseline	68.98	70.50	0.007

Table 4.8: A comparison between development models built using *CRF++* and *CRFSuite* and the top feature set for each POS model from the experiments in Section 4.2. The p-value is calculated using Wilcoxon signed-rank test.

A new experiment was set up to compare the best development performances achieved using *CRF++* models from Section 4.2 to *CRFSuite* counterparts with the new feature extractor, but using the same feature sets. The models were evaluated using ten-fold cross-validation. Table 4.8 compares mean development performance of each *CRF++* model with the respective *CRFSuite* model, showing that although *CRFsuite* achieves better scores in the majority of cases, the differences are not statistically significant for all but the baseline model comparison. It is therefore justified to use the faster *CRFSuite* library with a more flexible feature extractor for the remaining experiments.

5 The *CRFSuite* feature extractor source code, documentation, and installation instructions are available at https://github.com/savkov/CRFSuiteTagger

4.3.2 *Using The Universal Tagset*

The good results of the chunking model using a relatively simple part-of-speech annotation (*ARK-Twitter*) suggested that an even smaller tagset should be considered, which may be more fitting to the limited syntax of the primary care notes. In the general case, choosing or designing a tagset for POS tagging is mainly influenced by three factors — the purpose of the generated annotation, the amount of training data, and the distribution of targeted grammatical phenomena. Normally, POS tagging is used to support deeper linguistic analysis (parsing, NER, etc.), which may benefit from fine grained tagsets. However, annotated data used for training is limited and rarely has been compiled in a way that guarantees even representation of linguistic phenomena and types of language, which may lead to poor representation of less common part of speech types. The British National Corpus (BNC) was compiled with the aim of representing a wide range of variations and subdomains of British English, but only a small non-representative subset of its texts was manually annotated (Bentley et al., 1996). Possibly the most widely used resource in NLP, the Wall Street Journal part of the Penn Treebank (Marcus et al., 1993; Taylor et al., 2003), is comprised of narrow range of text types and subject domains.

The variety and scope of the language types used in these resources justified the use of fine-grained tagsets to support higher level analysis such as syntactic parsing. However, given the simpler global syntax structure of the Harvey Corpus text, and the results of the $\text{ARK}_{\text{Twitter}}$ model shown above, it was interesting to test if a simpler tagset was more suitable.

The Universal Tagset (UT), introduced by Petrov et al. (2012), aims to capture an optimal number of part-of-speech categories that are universal across languages in order to facilitate or avoid altogether the mapping of tagsets. The tagset has a number of potential applications, the most important of which is that it allows the building and evaluation of unsupervised and cross-lingual taggers and parsers. The difference between the language in primary care notes and the one in news articles can be seen as an extreme case of this particular application of the tagset, which makes it a good choice for these circumstances. It should be noted that the tagset is even smaller than the one used by the ARK tagger; although the missing tags are mostly specific to tweets, the proper name tag is merged into the noun tag, which could be an important difference.

There are two ways to test if another tagset improves chunking performance on primary care text. The naïve approach would simply map the existing annotations to the new tagset, and then use the data for training of chunking models. A more thorough approach would include mapping the original POS-tagging training data to the new tagset, and also re-training the tagging models. Under ideal circumstances the original text should be manually annotated in case the mappings are imperfect or not universally applicable to the data, however, within the scope of this thesis only the automatic mapping was considered.

	Original	**UT**	p	**Retrained**	p
ARK-NPS	77.52	77.12	0.139	-	-
ARK-Ritter	75.07	74.42	0.017	-	-
ARK-Twitter	78.17	77.95	0.103	-	-
cTAKES	76.82	76.64	0.202	-	-
GENIA	74.91	75.07	0.445	-	-
Stanford	76.54	76.42	0.683	75.87	0.012
SVMTool	75.41	75.14	0.541	74.68	0.017
Wapiti	75.11	74.72	0.203	74.01	0.009

Table 4.9: Comparison of chunking f-score between models using original POS annotation generated by the model, models using annotation converted to UT after POS tagging, and annotation generated by models trained using UT.

The chunking experiments were set up to test the two approaches described above. Unfortunately, not all models could be retrained due to lack of software functionality or unavailable data. Table 4.9 shows that simply converting the tagset in each version of the Harvey Corpus used for training did not yield positive results, compared to what was achieved with the original POS annotations. Even though the UT models have a lower accuracy, differences are not statistically significant in all but one of the cases. The results for the three models retrained using the UT show larger differences one of which (*Wapiti*) is statistically significant.

From the results of the conducted experiments it can be concluded that the effect of the universal tagset on chunking models is either negative or statistically insignificant. Therefore, it is justified to continue using the Penn Treebank tagset for the remaining experiments.

4.3.3 *Common Feature Types*

The main motivation behind using a more flexible tool and switching to the CRFSuite library was the opportunity to use a wider variety of features. Apart from word representation features there are a number of other feature types that might be appropriate, especially for sparse data with a considerable mixture of numbers, signs, and abbreviated words.

This section presents a series of experiments contributing to (in most cases significant) improvements in chunking performance. The baselines for comparison are models trained using the most successful feature set from Section 4.2. Section 4.3.3.1 explores the effects of different affix-based features, including these from the medical domain, Section 4.3.3.2 studies the effect of pre-processing and normalising the tokens before building the feature vectors, and finally Section 4.3.3.3 investigates n-gram-based features for words, as well as POS annotations.

4.3.3.1 *Affix Features*

The processing of morphologically rich languages often depends a great deal on the ability to preprocess and extract various morphological features. Such analysis can often be crucial for POS tagging and syntactic parsing. English has simple inflectional morphology, so affixes are normally not as important for NLP as they would be in many other languages, for example Finnish NLP. However, English has a rich derivational morphology, especially in forming nouns, which can be useful in guessing the part of speech of an unknown word. If we consider the made-up word *footion* without any context, the best guess that can be made about it is that it might be a noun, as it ends with the suffix *-tion*. In a nutshell, this is how morphological features help statistical classifiers — unknown words trigger very few features, so the ones that fire become much more important. Given the amount of terminology and spelling errors in primary care text, it is worth exploring whether morphology features could be of benefit.

There are two approaches to generating morphological features for machine learning — through manually created affix sets, and through automated extraction from text. The manually compiled affix lists can be divided into sub-types, e.g. based on their part of speech. For the purposes of these experiments, the following lists of affixes were compiled

from Wikipedia and Wiktionary[6]: *medical affixes*, *noun affixes*, *adjective affixes*, and *verb affixes*. The automated approach on the other hand, "generates" the affixes by cutting them out of a word during feature extraction using a predefined number of characters, e.g. the two-character suffix of *generation* is *-on*.

	Base	**Med**	*p*	**POS**	*p*	**All**	*p*
ARK_{NPS}	77.52	77.60	0.445	77.81	0.074	**77.93**	0.017
ARK_{Ritter}	75.07	75.37	0.169	75.44	0.059	**75.62**	0.059
$ARK_{Twitter}$	78.17	78.40	0.221	78.58	0.013	**78.73**	0.059
cTAKES	76.82	76.97	0.508	76.97	0.359	**77.09**	0.114
GENIA	74.91	75.43	0.074	75.15	0.508	**75.48**	0.074
RASP	76.68	76.63	0.575	**76.99**	0.169	76.96	0.203
Stanford	76.54	76.93	0.059	76.81	0.047	**76.88**	0.059
SVMTool	75.41	75.43	0.959	75.57	0.333	**75.83**	0.074
Wapiti	75.11	**75.46**	0.139	75.43	0.092	75.32	0.445
baseline	70.16	71.00	0.007	71.32	0.011	**72.13**	0.005

Table 4.10: Comparison between models without affix features (Base), and models with medical affixes features (Med), part-of-speech based features (POS), and the combination of the two (All). All *p*-columns refer to Wilcoxon significance tests performed between an affix feature model and the corresponding one without affix features. Top result indicated in bold.

MANUALLY-CREATED AFFIXES were grouped by location (prefix, suffix), and by origin (noun, adjective, verb, medical). A preliminary round of experiments showed that the verb affixes have a negative effect on the performance of chunking models, so they were not included in the following experiments. During those experiments the noun and adjective affixes were found to work best together, so they were bundled in the following comparison, that is the features were used together, but the affix groups (*medical suffix*, *noun prefix*) were kept separate in the feature vectors.

Three experiments were set up to test the effects of each group separately and together. Each was set up as a proof of concept experiment – new feature sets were tested using cross-validation on the development dataset as used previously for inner cross-validation. The context window of the new features was selected conservatively based on the most common token window among the best performing models from the experiments in Section

6 The scripts to scrape the suffixes are available at https://github.com/savkov/MedAffix

4.2 — [-1:0]. The results showed all affix types to make a slight improvement, although not statistically significant in all cases (see Table 4.10). Part-of-speech affix types in general made a greater contribution than medical affixes, but the models using the two together showed even better performance. While the differences between the different affix feature models are not statistically significant, many of them show notable improvements over the corresponding base models.

AUTOMATICALLY GENERATED AFFIXES have a few advantages over hand-crafted affix sets. They require no linguistic knowledge or manual work beyond a trivial implementation in a feature extractor. Additionally, they are representative of the training data, which means that they may generate more useful affixes even if they do not comply with morphological theory. However, they may generate misleading features that link very different words, e.g. *-oes* in *potatoes* and *goes*. Finally, there needs to be enough data to form well-defined classes in order to avoid noise.

POS	**M+P**	**Suffix**	*p*	**Prefix**	*p*	**All**	*p*
ark_irc	**77.93**	77.68	0.838	77.44	0.059	77.87	0.789
ark_ritter	**75.62**	75.44	0.308	75.14	0.059	75.59	0.878
ark_twitter	**78.73**	78.44	0.721	78.11	0.041	78.42	0.139
ctakes	77.09	77.33	0.241	76.79	0.241	**77.34**	0.367
genia	**75.48**	75.43	0.646	74.94	0.059	75.47	0.878
rasp	76.96	**77.22**	0.260	76.69	0.333	77.06	0.485
stanford	**76.88**	76.82	0.575	76.62	0.333	76.66	0.333
svmt	**75.83**	75.65	0.575	75.19	0.059	75.41	0.386
wapiti	**75.32**	75.28	0.508	75.06	0.139	75.33	0.951
Baseline	72.13	72.41	0.291	70.50	0.007	**72.63**	0.176

Table 4.11: Comparison between models with tailored affix features (M+P), automatically generated suffixes (Suffix) and prefixes (Prefix), and models with manually-crafted affixes and automatically generated suffixes (All). All *p*-columns refer to Wilcoxon significance tests performed between a model with automatically generated affix features and the corresponding model with hand-crafted affix features. Top result indicated in bold.

Three experiments involving automatically generated three-character affix features were set up, and their results were compared to the models using only hand-crafted affixes. Table 4.11 shows that prefix features have a negative effect on chunking performance, while

models with suffix features score slightly lower than the optimal combination of manually-created affix features (although not with a statistically significant difference). The combination of manually-crafted affixes and automatically generated affixes yields slightly better results than only automatic suffixes and slightly worse than manually-created affixes. No statistically significant improvement in accuracy was noted beyond that of the models with hand-crafted features. The same set of experiments was also conducted with a two-character affixes, but the results were similar. Therefore, even though automatically generated affixes contribute to the accuracy of the baseline model, they have a lesser impact than manually-created features when used in conjunction with POS annotation.

4.3.3.2 *Canonicalisation Features*

One of the weaknesses of token features is the presence of a large proportion of singleton tokens. Many numbers and misspelled words occur only once or twice in the Harvey Corpus. At this level such features are regarded as uninformative and ignored. Most machine learning tools and libraries have a minimum feature occurrence threshold. In the case of misspelled words that is probably the best that can be done, short of correcting the mistakes. However, in the case of numbers, valuable information can be gathered if all unique number tokens are all treated as a single numeric token. Feature canonicalisation takes tokens with varying form, but a common semantic type, and replaces them a place holder token, e.g. <NUMBER> for any number expressed with digits.

The Harvey Corpus is rich in numbers and quantities, but it also has a specific feature that fits canonicalisation perfectly, namely redacted text. Redacted text as discussed previously is represented by a string of tilde characters of equal number to the letters in the redacted word. Given that many of the redacted tokens were names of people or places, replacing the tildes with one common token that behaves like a name could be considered as extracting hidden information from the data.

The experiments were conducted in the same manner as the previously described affix features experiments using cross-validation and a narrow context window. The results in Table 4.12 show that the models with canonicalisation features achieve significantly better results than the baseline and the affix feature models. The combination of canonicalisation and affix features, however, does not seem to make a statistically significant difference in

	Baseline	A	C	p	A+C	p
ARK_{NPS}	77.52	77.93	78.12	0.007	**78.70**	0.007
ARK_{Ritter}	75.07	75.62	75.51	0.139	**76.15**	0.139
$ARK_{Twitter}$	78.17	78.73	78.62	0.007	**79.18**	0.007
cTAKES	76.82	77.09	77.38	0.760	**77.50**	0.760
GENIA	74.91	75.48	75.63	0.103	**76.02**	0.103
RASP	76.68	76.96	77.13	0.074	**77.46**	0.074
Stanford	76.54	76.88	77.61	0.721	**77.68**	0.721
SVMTool	75.41	75.83	76.33	0.333	**76.64**	0.333
Wapiti	75.11	75.32	75.67	0.445	**75.84**	0.445
baseline	70.16	72.13	71.78	0.009	**73.26**	0.009

Table 4.12: A comparison between models with affix features (A), models with canonicalisation features (C), and models with both (A+C). The optimal configuration achieved during the Section 4.2 experiments used as the baseline. Significance tests are for the pairs A:C and C:A+C using a Wilcoxon test. Top result indicated in bold.

most cases. However, it should be noted that in the cases where such a difference exists it is with a very low p-value.

4.3.3.3 *N-gram Features*

All the context feature discussed so far are sequences of length one, i.e. unigrams. Bigrams and other longer sequences have more information content than unigrams, however, as features they are not always preferred over them. Longer n-grams are generally used when there is a large amount of data, enough to ensure that enough combinations of items will occur more than just once or twice. Part-of-speech n-grams can become useful with less available data as the number of unique items is small compared to word n-grams.

As discussed previously in Chapter 3, the Harvey Corpus is not particularly large. Therefore it is unlikely that n-gram token features will have a positive influence on chunking models, but POS n-grams are worth investigating.

The experiments presented in this section were designed to determine if token and part-of-speech bigram features have a positive effect on chunking. As a result a series of experiments were conducted, three of which are presented in Table 4.13.

Token bigrams were not expected to contribute to the chunking performance given the small size of the corpus, but the experiments showed that, in fact, in many cases they even have a negative effect. In contrast, part-of-speech bigrams show improvement both

	Base	A+C	N_t	N_p	$N_{t,p}$+A+C	p
ARK_{NPS}	77.52	78.70	76.72	77.71	**78.97**	0.126
ARK_{Ritter}	75.07	76.15	74.68	75.80	**76.52**	0.445
$ARK_{Twitter}$	78.17	79.18	77.37	78.37	**79.31**	0.374
cTAKES	76.82	77.50	76.02	77.33	**78.05**	0.059
GENIA	74.91	76.02	74.92	76.09	**76.93**	0.013
RASP	76.68	77.46	76.08	76.83	**77.77**	0.386
Stanford	76.54	77.68	76.50	77.45	**78.42**	0.017
SVMTool	75.41	76.64	75.23	76.24	**77.42**	0.013
Wapiti	75.11	75.84	74.57	75.69	**76.54**	0.028
baseline	70.16	73.26	67.53	70.42	**74.17**	0.009

Table 4.13: A comparison between a model using affix and canonicalisation features (A+C), a model with token bigram features (N_t), one with POS bigram features (N_p), and a model with the combination of them all ($N_{t,p}$+A+C). The p-values were calculated for the differences between A+C and $N_{t,p}$+A+C. Top result indicated in bold.

when used alone and in combination with the affix and canonicalisation features. The results of the bigrams alone are not better than what was achieved in the experiments from the previous subsection. The combination of bigrams and the best configurations so far increases the performance of the models, but not by a statistically significant margin (see the p-value column of Table 4.13).

A further experiment with part-of-speech trigrams was conducted, but it showed slightly lower and statistically insignificant performance compared to the combination of bigram, affixes, and canonicalisation features.

4.3.4 *Word Representation & Clinical Text*

Semantic word representations have been investigated intensively in NLP research in recent years. They allow machine learning tools to harness the knowledge from large amounts of raw unannotated text. Four of the POS-tagging models use clusters generated based on the context distribution of words. The ARK tagger (Owoputi et al., 2013) uses a clustering of 216 thousand word types (unique tokens) distributed in 1,000 clusters using Brown hierarchical clustering (Brown et al., 1992). The clusters were generated from a corpus of 847 million tokens. The Stanford NLP package (Manning, 2011) uses Brown clustering as described and implemented by Clark (2013) generated from the Reuters RCV1 (Lewis

et al., 2004) and Gigaword 4 (Napoles et al., 2012) corpora. Word embeddings have also been applied to biomedical NLP (Stenetorp et al., 2012b).

Generating such clusters and embeddings can be a long, laborious, and computationally challenging process, so a certain set of pre-computed clusters and embeddings are commonly used by researchers. The experiments in this section are separated into two groups according to the origin of the resources they use. First, in Section 4.3.4.1, publically available general domain clusters and embeddings are tested through evaluation of chunking models enhanced with features generated from them. Cluster features were represented as an integer denoting the cluster number, while the word embeddings were simply concatenated (in the same position) to the existing feature vector. Then Section 4.3.4.2 discusses the effect of using clusters and embeddings generated from biomedical and clinical text.

In order to keep the number of experiments within bounds, only four POS models were selected for testing. From the results presented so far it is clear that the $ARK_{Twitter}$, ARK_{NPS}, and the *Stanford* annotations lead to the best performing models, so they were selected along with the baseline — no POS annotation. The style of the experiments was slightly changed from the previous proof-of-concept experiments to set the context window for word representation features to only cover the focus token ([0]). The latter limit was adopted, because the preliminary experiments showed that larger feature windows lead to decreased performance. The small amount of both labelled and unlabelled data that the features and classifiers use is a very likely reason for this decrease. However, analysing the exact cause is outside of the scope of this thesis due to the limited amount of domain data.

4.3.4.1 *Pre-computed Clusters & Embeddings*

The amount of data used for generating word representation embeddings and clusters is one of the most crucial factors for their success. However, the greater the size of the used data, the more time and resources (and skills to some extent) are required to perform the computation. Researchers often make such resources publicly available as they have no legal restrictions like corpora, and some have become a baseline for comparisons in the NLP field. Stenetorp et al. (2012b) compare some "standard" clusters and embeddings to ones created from biomedical text when used in biomedical NER and semantic category disambiguation. They concluded that in-domain word representation features show greater and more consistent benefits than those based on out-of-domain (general purpose) data.

This section examines the effectiveness of publicly-available embeddings and clusters on chunking evaluated on the Harvey Corpus. The Collobert and Weston (2008) style embeddings and Hierarchical Log-Bilinear (HLBL) embeddings (Mnih and Hinton, 2008), as well as RCV1 Brown (Brown et al., 1992) clusters, all generated as part of experiments by Turian et al. (2010), were acquired from the *MetaOptimize* web page[7]. Also acquired was a set of word embeddings generated by Google using *word2vec* (Mikolov et al., 2013) on a Google News corpus with nearly 3 million word types[8]. Another three cluster resources were also acquired: the in-domain Brown clusters generated by Stenetorp et al. (2012b)[9], the Ney-Essen clusters (Ney et al., 1994) used by the Stanford NLP package generated by Clark (2003)[10], and the Brown clusters used by the ARK POS tagger (Owoputi et al., 2013)[11].

type	**size**	**ARK$_{\text{Twitter}}$**	**ARK$_{\text{NPS}}$**	**Stanford**	**No POS**
CW	100	79.04	78.38	77.23	73.40
CW	200	79.18	78.25	77.33	73.50
CW	25	79.30	78.57	77.42	73.37
CW	50	79.16	78.48	77.29	73.42
OSCCA	200	79.22	78.35	77.10	74.13
TSCCA	200	79.22	78.90	77.53	**74.67**
HLBL	100	**79.39**	78.46	77.28	73.53
HLBL	50	79.05	78.33	77.30	73.52
word2vec	300	79.31	78.49	77.54	73.31
baseline		79.31	78.97	78.42	74.17

Table 4.14: Comparison of chunking models using pre-computed word embeddings (vectors). CW = Collobert and Weston (2008) embeddings; *CCA = Canonical Correlation Analysis embeddings (Dhillon et al., 2015); HLBL = Hierarchical Log-Bilinear embeddings (Mnih and Hinton, 2008). The baseline is the best performance in experiments described so far without features based on embeddings. The size is the vector size of the embeddings. Top result indicated in bold.

The series of experiments evaluating the effect of features based on the embedding resources showed that they do not improve chunking performance in almost all cases. There is only one case of significantly better performance (p-value at 0.1): TSCCA embedding fea-

7 http://metaoptimize.com/projects/wordreprs/
8 https://code.google.com/p/word2vec/
9 http://wordreprs.nlplab.org/
10 http://nlp.stanford.edu/software/egw4-reut.512.clusters
11 http://www.ark.cs.cmu.edu/TweetNLP/clusters/50mpaths2

tures help improve the performance of the model without POS features. In contrast there are a number of cases where there is a statistically significant decrease in the performance of the *Stanford* annotation models (see Table D.1). The lack of statistically significant difference in most of the experiments with a tendency for negative impact in the rest suggests that the embeddings add little useful information beyond what is already there. There are three factors that could be the cause of such an outcome: limited word coverage, limited impact (features do not change anything), and unfavourable impact on the feature vectors. The vocabulary of the embeddings covers 79.8% of the tokens and 62.9% of the word types in the cross-validation set, which is a reasonable level of coverage. The output of models with embedding features was compared to the output of the same models without embedding features, and it was found that the number of affected tokens ranges between 2.5% and 3.5% — more tokens were affected in models without POS features. Furthermore the numbers of tokens with improving and deteriorating chunk annotation is slightly in favour of deterioration. Therefore, given that there is generally little change in performance, and even a few cases of decline, while there is significant impact on the classification process, it can be deduced that the information introduced by the embeddings is of little use. That may be due to the cross-domain usage of the embeddings, but it is also possible that the baseline feature set has a very similar effect thereby cancelling the improvement. The second possibility was rejected through a series of experiments where the effects of the embeddings were measured on models with minimalistic feature sets, and also found to be statistically insignificant.

The cluster features, the other form of word representation features, were tested in the same type of experiments as word embeddings. The results in Table 4.15 single out the *Ney-Essen* clustering as giving the best all-round performance. Although all other clusters made significant improvements to models without POS features, it is particularly interesting that the Ney-Essen clusters managed to increase it by nearly 3 percentage points. Another interesting issue is the performance of Brown cluster features based on the work of Stenetorp et al. (2012b) which were created from PubMed abstracts. Although in some cases there are significant differences from the baseline (e.g. 320 and 500 PubMed models), the two groups of clusters show no statistical significance.

In summary, word embeddings generated from out-of-domain data are of very little use when applied to this primary care chunking task, but clustering features show significant

type	source	size	Twitter	NPS	Stanford	No POS
Brown	Twitter	1,000	79.28	78.60	77.85	76.28
Brown	PubMed	100	79.46	78.87	77.84	76.39
Brown	PubMed	1,000	79.52	78.85	78.07	75.96
Brown	PubMed	150	79.44	78.66	77.72	76.25
Brown	PubMed	320	**79.71**	78.75	78.15	75.84
Brown	PubMed	500	79.68	78.53	77.87	75.93
Brown	RCV1	100	79.42	78.65	78.02	76.26
Brown	RCV1	1,000	79.34	78.59	77.91	75.96
Brown	RCV1	320	79.47	78.80	77.67	76.18
Brown	RCV1	3,200	79.32	78.44	77.45	75.97
Ney-Essen	RCV1	512	79.26	**79.39**	**79.11**	**77.08**
baseline	-	-	79.31	78.97	78.42	74.17

Table 4.15: Comparison of chunking models using pre-computed cluster features. Top result indicated in bold.

contribution in all cases of missing POS features, as well as some of the others. This suggests that cluster features, especially if well designed, could replace a large part of the contribution of POS features.

4.3.4.2 *In-domain Clusters & Embeddings*

The type and amount of the training data are probably the most important factors in applying machine learning to any problem. Stenetorp et al. (2012b) show that the choice of in-domain data for word representation cluster features has a positive effect on NER performance for biomedical text (PubMed abstracts). However, in the previous subsection it was shown that the same word embeddings show very little difference when applied to primary care text. As there are no publically available embeddings or clusters based on primary care data, the only way to determine if in-domain data is helpful was to generate them from the available GPRD data. The total amount of unlabelled data including the Harvey Corpus (without the development and test sets) amounts to approximately 650,000 tokens, and 44,000 word types. Although the amount of tokens in the data is half as much as the Wall Street Journal part of the Penn Treebank (which has roughly 1.28 million tokens, 51,500 word types), it is tiny compared to the typical size of corpora commonly used for generation of embeddings and clusters, such as *RCV1*, *English Wikipedia*.

	size	$\text{ARK}_{\text{Twitter}}$	ARK_{NPS}	Stanford	No POS
word2vec	100	79.27	78.47	77.38	73.46
word2vec	25	78.93	**78.72**	**77.48**	73.33
word2vec	50	**79.43**	78.51	77.39	**73.69**
baseline		79.31	78.97	78.42	74.17

Table 4.16: Chunking performance of models with *word2vec* embeddings generated from GPRD data. Top result indicated in bold.

Two groups of experiments analogous to the ones described in the previous subsection were conducted to test the effects of in-domain embeddings and clusters on the chunking of primary care text. In preparation, a set of word embeddings were generated from the GPRD data using *word2vec*, as well as two types of clusters of different sizes — Brown and Ney-Essen. The clusters were generated using the implementations by Liang (2005)[12] and Clark (2003)[13] respectively.

The results of the new word embedding experiments showed a similar pattern as pre-computed word embeddings. No significant improvement was achieved, and in a few cases performance even decreased significantly (see Table 4.16). In contrast, the cluster features led either to no significant change or to improvement in performance. The trend in improving the *No POS* models is still present, although not by the same margin. It is interesting to note that the 250 Brown clusters achieved the best performance so far in these series of experiments, although not significantly better than the results of other models using pre-computed word clusters.

Clustering	size	Twitter	NPS	Stanford	No POS
Brown	100	79.36	79.02	77.99	75.25
Brown	250	**79.81**	**79.03**	77.92	75.55
Brown	500	79.49	78.87	77.94	75.60
Ney-Essen	500	79.21	79.00	**78.67**	**75.86**
Brown+NE	250 & 500	78.06	77.09	76.84	72.02
baseline	-	79.31	78.97	78.42	74.17

Table 4.17: Chunking performance by models using Brown and Ney-Essen cluster features generated from the GPRD data. Top result indicated in bold.

12 https://github.com/percyliang/brown-cluster
13 https://github.com/ninjin/clark_pos_induction

The results of the experiments both using pre-computed and in-domain embeddings and clusters can be summarised in the following way. First, it is evident that embedding features have mostly no significant effect, and sometimes even a negative influence on chunking performance on this primary care text. Second, cluster features show both positive and negative influence by a good margin, but the Ney-Essen clusters trained on RCV1 show a notable improvement in three out of four tested models. And third, even though models with in-domain word clusters achieved significant improvements compared to the baseline, they fail to surpass their pre-computed generic counterparts. In practical terms, it seems that Ney-Essen clusters are a poor choice for models with *Twitter* POS annotation, while they provide an impressive performance boost to both *Stanford* and *No POS* models.

Judging from the results, Brown and Ney-Essen clusters seem to complement each other across different POS annotation models, so it was worth investigating if using them together would work better than using them separately. An experiment was set up using Brown and Ney-Essen clusters from GPRD data. The 250 Brown clusters were used since they were the highest ranking of the three Brown clusterings. The results showed a significant decrease in the performance of all models.

4.4 SOLVING A COMPLEX PARAMETER TUNING PROBLEM

Finding informative types of context features is only a part of shaping the optimal feature set for a given task. The performance of each feature type is largely influenced by the scope of its context: if the scope is too wide, it may introduce noise and bring down performance, whereas if it is too narrow its effect may be marginal. Finding the optimal scope of a single feature empirically is usually not difficult, given sufficient parallel processing resources. However, the informativeness of machine learning features also depends on other features present in the feature vector. Thus the size of the optimisation problem grows exponentially with the the number of features in the vector:

$$s = (w_{\mathtt{left}} \cdot w_{\mathtt{right}})^n \tag{4.1}$$

The w's are the sizes of the explored context window on each side, and n is the number of context features. Therefore, exploring the full search space of context windows for any feature vector with more than a few features becomes intractable. For instance, a grid search of all context window combinations of the feature vector used in the experiments described in Section 4.3.4 would give a problem size of 4.9517602×10^{27}.

The rest of this section explores two approaches to feature optimisation, compared to a baseline of greedy feature-by-feature optimisation. Bayesian Optimisation (BO) is discussed in Section 4.4.1; this is a method that employs machine learning to make a series of tests in order to determine the global optimum. A slightly less greedy version of the baseline optimisation method is described in Section 4.4.2.

All experiments considered below are based on data using the *ARK-Twitter* POS annotation, and the list of useful features excluding word representation (i.e. after experiments in Section 4.3.3). The baseline method optimises the model by adding and optimising each feature in a subjective order of importance. This method achieved 78.16, which is significantly lower than the f-score reported in Section 4.3.3.3.

4.4.1 *Bayesian Optimisation*

One of the most common problems when working with machine learning classifiers is hyperparameter tuning. Typically this task is carried out either based on the researcher's intuition about good parameter ranges, or brute force grid search of a range of values. Needless to say the first approach is far from exhaustive, while the second one is computationally expensive and offers limited guarantees of success. Bayesian optimisation (Mockus, 1977) optimises the parameter $x \in R^D$ of a function $f(x)$ on some bounded set X. The feature that sets Bayesian optimisation apart from other methods is the usage of a probabilistic model based on Bayes' law to infer the parameter values for every further evaluation of $f(x)$. The advantage of this approach is that it uses all the information generated from previous evaluations of $f(x)$ to significantly decrease the number of iterations needed to find the minimum of a non-convex function. Bayesian optimisation computation comes at a greater cost compared to other optimisation techniques, but in the case of optimising the parameters of machine learning models, may be justified since re-training a model is often expensive.

Although the typical application of Bayesian optimisation is tuning to hyperparameters, one can also use it to tune context windows in feature vectors. For example, tuning the context window of a feature can be represented as tuning a two parameter function, the parameters being the borders of the context window and the function being the evaluation of the model using this defined context window. Snoek et al. (2012) describe an algorithm implemented in the Spearmint system that can surpass a human expert level of optimisation of a variety of machine learning algorithms. The algorithm can also surpass human expert optimisation of functions (algorithms) with four parameters. Furthermore, it allows parallel experimentation that can take advantage of multiple processing cores[14].

Even though the algorithm has been shown to deal with optimisation of multiple parameters, it seems unlikely that this would be as successful when tuning whole feature vectors such as the ones discussed in Section 4.3.4, given their size (more than ten feature types). An initial experiment based on the feature vector of the best performing *ARK-Twitter* model without word representation was carried out to get an indication of whether the algorithm could achieve the same performance using Bayesian optimisation (BO). The initial test was set to optimise four features at the same time (tokens, POS tags, POS tag bigrams, and canonicalisation) in order to maximise the chances of finding a maximum within the first few iterations (with the limit set to 50 iterations for this experiment). The rest of the context features were kept as they were. The optimised parameters were set to be integers in the ranges [-3:0] and [0:3]. The best performance of the algorithm was obtained in the 42nd iteration, when the chunker achieved an f-score of 78.83, which is higher than the baseline (p-value 0.053), but significantly lower than what was achieved during the previous experiments (79.31). The experiment was extended to more than 500 iterations and the same score was achieved again, but never surpassed. It should also be noted that the configuration suggested by the algorithm differed fundamentally, as it was using a much wider context for all but the canonicalisation features.

It was unclear whether the shortcomings of the algorithm are caused by the number of optimised parameters or by much more sensitive relationships between the tuned parameters. Therefore a further experiment was carried out which instead of optimising eight parameters at the same time, separated the parameters into two groups, which were optimised in sequence — the second group building on top of what was achieved by the

14 A Python implementation of Snoek et al.'s variation of Bayesian optimisation is available at https://github.com/HIPS/Spearmint

first. The groups were formed by ordering the features by intuitive importance (see the order in the final feature vector in Table C.2) and putting the more important ones in the first group and the less important ones in the second group. The resulting configuration, found in 22 + 14 iterations matched exactly the one from the manually crafted configurations. This achievement showed that Bayesian optimisation can deal with the optimisation problem in smaller chunks faster than a grid search, which would require 256 iterations to tune two pairs of features; however optimising more parameters at a time comes at a performance cost. It should be noted that BO does not complete upon finding a local or global optimum, but upon a pre-set number of iterations. Thus a more realistic appraisal of its effectiveness should consider a fixed number of iterations needed to find an optimum, e.g. 50 or 100, rather than stopping at an optimum. Considering an approximate cost of 50 iterations for BO (100 in total), the cost difference between using BO and grid search in the same way is only 2:5. Given the capabilities of current high performance computing clusters, BO's modest reduction in processing cost is not worth the risk of missing the global optimum.

4.4.2 *Greedy Parameter Group Optimisation*

The baseline method described above takes a computationally cheap, but also naïve approach to feature vector optimisation. It assumes a feature modularity that does not exist in reality — the gain yielded by two features together is not necessarily equal to what is gained from them being used separately. In addition, due to optimising features one at a time, the baseline assumes that the order in which features are added and optimised is the most suitable, although there is no hard evidence supporting that assumption.

So the baseline makes assumptions that are very likely to be misleading, but as previously explained, if all features are optimised simultaneously the process becomes intractable. However, the possibility of inaccuracies due to the assumptions can be decreased if features are optimised in groups of three instead of one at a time. This means that during the development process features are added in triples and optimising after each vector extension. In this case the choice of the sequence of optimisation is less important as more features are optimised together, but an element of a human "craft" still exists (unless, of course, all possible permutations of groups are tested). The difference between the baseline

and the results achieved in Section 4.3.3, where new features are added to a fully optimised group, already shows that group optimisation works better.

When applied to the feature vector from Section 4.3.3.3 using the *ARK-Twitter* data, the greedy group optimisation method yielded a slight f-score improvement reaching 79.36, which is of no statistical significance. The feature vector resulting from the optimisation differs only marginally from the narrow context windows used in the initial experiments — only one feature context border is different. This shows that the intuition of using a narrow context window for the feature vector development was a good idea. However, out of the considered optimisation methods, greedy group optimisation is the best compromise between performance and reliability.

4.5 FINAL OPTIMISATION

Although the experiments discussed so far give a good idea about which features should be used in the optimal feature vector, it is difficult to make an objective choice of clustering features for a single "final" optimisation. Instead, one of each group of clustering features was chosen to be optimised using greedy feature group optimisation based on their average performance across the evaluated models. Groups were defined by the combination of clustering method and source data. Thus both Ney-Essen instances were included, as well as Brown clusters from RCV1, PubMed, and GPRD data. The number of clusters selected in the latter three were 320, 320, and 250 respectively. The results of this optimisation experiment were used to determine the model used for further experimentation in the thesis. It is important to choose only one model during development before evaluating on the test set, since the concept recognition experiments intended to benefit from the model (these are described in Chapter 5) had to use the same data splits. Thus if a model had been chosen based on test set performance, this would have introduced a bias in the concept recognition experiments.

The parameter groups optimisation yielded little improvement over the proof-of-concept experiments, at least for the majority of the models (see Table 4.18). A noteworthy exception is the Ney-Essen clustering based on the GPRD data, which has improved considerably (p=0.036). The overall experiment results offer clear evidence that the $ARK_{Twitter}$ models performs better than the *Stanford* models ($0.01<p<0.05$ for all relevant p-values), and

	NE-R	NE-GP	B-PM	B-GP	B-R
Twitter	79.49	**79.97**	79.82	79.76	79.48
NPS	79.27	**79.32**	79.18	79.30	78.80
Stanford	**79.11**	79.03	78.35	78.56	78.04

Table 4.18: Final optimisation results for models with Ney-Essen (NE) and Brown (B) clustering, using data from the GPRD (GP), PubMed abstracts (PM), and Reuters Corpus (R).

marginally better than the ARK_{NPS} ones ($0.01<p<0.10$ for all relevant p-values). Considering only the *Twitter* models, the clusters based on RCV1 perform marginally worse than those generated from GPRD and PubMed data. The search for the model with the optimal performance eventually came down to a choice between three models: Ney-Essen clusters from GPRD data (NE-GP), Brown clusters from PubMed data (B-PM), and Brown clusters from GPRD data (B-GP). The lack of a statistically significant difference makes the choice of the highest scoring model among them rather arbitrary, yet the NE-GP model was *selected* based on the f-score as "most optimal" for the lack of a better motivation (see the feature vector template used for the model in Table C.2).

Upon the completion of the development process a final experiment using the final feature vector was carried out to determine the model performance on the test set. As previously stated, for the final evaluation the model was trained on the full development set (90% of the whole corpus) and evaluated on the test set (the remaining 10%). It achieved a slight (statistically insignificant) increase in the performance, reaching an f-score of 79.89. This result was a sign of successful model development, since the final model evaluation showed a performance consistent with the results from the development process.

4.6 CHAPTER SUMMARY

This chapter provided an account of the development of machine learning models for chunking, optimised for the UK primary care data of the Harvey Corpus. Initial experiments showed that pre-existing part-of-speech tagging tools and models achieve reasonable accuracy, whereas pre-existing chunking models have poor accuracy. Models trained on the Harvey Corpus using POS annotation generated by various publicly available taggers

showed a performance that is much better, although far below the best results reported for news text or PubMed abstracts (see Section 4.1).

Section 4.2 established an experimental baseline using a data split allowing model development using 10-fold cross-validation over the development set (90% of the whole corpus), while performing the final evaluation on the test set (the remaining 10%). Optimisation experiments were conducted to determine which machine learning tool performed better, CRF++ or YamCha (Section 4.2.2), and what chunk representation is most suitable for the data (Section 4.2.3). The results showed that CRF++ yields significantly higher accuracies than the SVM-based YamCha. In contrast, the two chunk representation schemes considered showed no statistically significant difference, although BEISO achieved higher scores in the majority of cases. However, the commonly used BIO was selected as being more comparable with other research in the field (as used in the CoNLL-2000 shared task in chunking). Finally, the experiments indicated that differences in POS annotation are a fundamental factor in developing a chunking model, but it is difficult to determine a single most favourable POS annotation model. Nevertheless a group of POS annotations generated by the $ARK_{Twitter}$, ARK_{NPS}, and *Stanford* POS models were considered to be the top candidates, closely followed by those of *cTAKES* and *RASP*.

Section 4.3 describes a series of further experiments that were conducted using the same experimental scheme in order to test a number of additional features. A new feature extractor wrapped around the CRFSuite library was developed to allow models with more flexible feature types to be developed (Section 4.3.1).

The hypothesis that a simpler tagset would improve performance was tested in Section 4.3.2 using the universal tagset as there are pre-existing mappings from it to other popular tagsets. The experiments rejected the hypothesis, both when the tagset was directly substituted in annotated Harvey data, and when retrained UT models were used to annotate the Harvey Corpus. One likely reason for the lower performance, compared to the ARK tagset, is that the proper name tag, used by ARK, is merged with the other noun-like tags into the N tag in UT. This affects over 2000 occurrences, which is roughly 7% of the tokens in the corpus. Designing a new similar tagset with consideration for the data might be a potentially good and interesting idea, which can be pursued in future work.

Three types of well-established features were tested and found beneficial for chunking models in Section 4.3.3. Medical and POS based affix features were introduced, as well as

POS n-grams and canonicalisation features. A wide range of features based on semantic word representation embeddings and clusters were also evaluated. None of the embedding features were found to contribute substantially, while word representation clusterings had mixed success depending on the data they were based on. The Ney-Essen clusters showed the most consistent improvement, as well as the highest gain over the baseline model (without POS annotation). Brown and Ney-Essen clusters generated from the available GPRD data also showed significant improvements to the baseline, and in some of the other models. The highest development f-score was achieved by a model using Brown clusters (250) based on the GPRD data.

Section 4.4 discusses two additional feature vector optimisation techniques: Bayesian optimisation including all feature parameters and a greedy group optimisation. These are considered in addition to the greedy one-by-one optimisation which was essentially implemented during the feature vector development process presented in Sections 4.2 & 4.3. None of the considered optimisation methods showed a statistically significant improvement. However, BO still poses some small risk of reaching a local maximum, while the greedy group optimisation offers comparable reduction of the experiment rounds avoiding that risk.

Finally, Section 4.5 describes the final optimisation using the greedy parameter group optimisation approach. The model groups were defined by the type of word representation cluster features they used. The optimisation suggested that the $ARK_{Twitter}$ models had better performance than the other two POS models considered. Models with cluster features based on RCV1 performed marginally worse than the ones based on PubMed and GPRD. The model using Ney-Essen clusters based on GPRD data was selected as the best out of the three highest performing models based on its higher performance. When evaluated on the test set, it achieved 79.89 f-score, which is consistent with its results on the development set, indicating a robust model development process.

The accuracy reported in the final chunking evaluation of this chapter is considerably better than the first experiments that used knowledge and resources adapted from another domain. However, in absolute terms it is still quite far from the results achieved by the same and similar approaches on the Penn Treebank, biomedical abstracts, and even some types of clinical data. However, one can argue that the state of the art in other domains is not the correct benchmark. Instead, in-domain human consistency should be considered

as what can be realistically achieved. If the inter-annotator agreement reported on the Harvey Corpus in Chapter 3 is considered as the goal, then the best performing model is just over five percentage points short of achieving it.

In conclusion, this chapter presented the development of a text chunker for the Harvey Corpus, working in conjunction with a pre-trained part-of-speech model originally developed for tagging tweets. Together, the two models provide a solid foundation for further analysis, working to a level of performance that is reasonable given the characteristics of the data.

5

MEDICAL CONCEPT RECOGNITION

Recognising semantic entities is an important processing step for many natural language processing applications. It is a broadly defined task since what classes should be recognised (whether abstract ones or such referring to entities in the real world) depends on the application. Colours, shoe brands, feelings, and chemical compounds are all valid semantic classes. Not all such classes need a complex algorithm to be recognised — searching for keywords or simple patterns may be sufficient to recognise some, while others could be so abstract that none of the current technology could recognise them. Their content has very few restrictions by definition — any sub-sentence sequence of tokens can be an entity, as long as its meaning fits the class definition. The same rationale can be applied to medical concepts such as symptoms, diseases, and drug names, the recognition of which is the subject of this chapter.

Named entities are the most commonly targeted type of semantic entities, and the process of their recognition is an established NLP task. Its goal is to identify and classify any rigid designators, i.e. names, that reference specific entities, such as people, locations, etc. Temporal expressions and several other number-oriented semantic entities have also been considered to be part of the task, even though they are defined in a completely different way. In clinical NLP, medical concepts are an important kind of semantic entities (Uzuner et al., 2010a, 2011; Bada et al., 2012) due to their paramount importance to understanding clinical data.

As part of PREP and using one of the datasets used for the Harvey Corpus, Tate et al. (2009) showed that many symptoms commonly associated with ovarian cancer are recorded in the text part of medical records prior to the recorded date of diagnosis: abdominal pain (41%), urogenital problems (25%), abdominal distension (24%), constipation/change in

bowel habits (23%) with 70% of cases reporting at least one of these. They concluded that "[f]ree text information may be essential in obtaining accurate estimates of incidence, and for accurate dating of diagnoses." Similar conclusions were reached by Ford et al. (2013) using PREP data for rheumatoid arthritis patients. Koeling et al. (2011b) reported that incidence estimates of ovarian cancer symptoms increased by at least 40% when complementing coded (structured) data with symptoms manually extracted from free text.

Given the potential benefits of including information from free text notes in epidemiological studies, the automatic detection of symptoms, drugs, and diseases becomes an important step towards enhancing future research involving primary care data. The semantic entities discussed in this chapter are medical concepts roughly defined as *diseases*, *symptoms*, and *drugs*. They are typically confined in noun phrases, so they are likely to be only partially affected by the terse language style of primary care text, while their recognition should be aided by the existing noun chunk annotation. The importance of noun chunks is also the reason the annotation and modelling of these entities was packed in a separate task. A system able to automatically recognise these three types of concepts in primary care text could be of great use to epidemiologists and other e-health researchers and data scientists trying to explore massive quantities of clinical data.

The rest presented below describes the process of developing a machine learning model for concept recognition based on the Harvey Corpus. Section 5.1 recounts the process of designing and implementing an extension to the Harvey Corpus annotation with the concepts listed above. Section 5.2 and 5.3 present two approaches to the task: the first is the usual approach for BIO annotation tasks, which uses sequential taggers, while the second utilises the particular language characteristics of primary care text to make the assumption that all medical concepts are base NPs, which allows the task to be formulated as document classification. Finally, Section 5.4 compares the two methods in a realistic setting.

5.1 EXTENDING THE HARVEY CORPUS

Semantic annotation is in general a difficult undertaking due to the somewhat looser definitions of entities compared to parts of speech for example. Another difficulty arises in annotating higher level semantic entities that may not be confined to a single syntactic

constituent or a sentence. This makes difficult to formulate precise annotation guidelines, and shortcomings are inevitable in certain cases. However, the extent to which the annotation task is affected by these shortcomings depends on the nature of the targeted entities. Classic NER, for example, is less affected by it, because it focuses mainly on measurement units, certain quantifiers, and names, which in general have clearly identifiable borders in the text, at least in the majority of cases. The observations about inter-annotator agreement of semantic entities made in Chapter 3 demonstrate that the task difficulty is largely dependent on the type of entities being annotated. Annotators agreed upon time expressions much more often than upon locative expressions. Often the most difficult part of annotating such entities is not so much detecting their presence, as recognising their correct boundaries. This section describes the annotation of symptoms, diseases, and drug names, which circumvents boundary recognition issues by using the boundaries of existing NP annotation to define units for the annotation.

5.1.1 *Annotation Approach*

The telegraphic, NP-heavy style of the primary care notes of the Harvey Corpus puts the design of a semantic annotation scheme at an advantage. The incentive of medical workers to pack as much thought as possible in as little text as possible has its consequences in the expression of medical concepts as well as syntactic structures. As previously stated, when looking at the data, it is easy to notice that medical concepts, such as symptoms, diseases, and drug names, are almost always expressed as a single base NP. A simple assumption that this observation is true in all cases would make the annotation task much simpler and safer (from an agreement point of view), as it explicitly defines the units to be annotated (i.e. classified), rather than relying on the annotators to consistently find their boundaries in the text. In a way, this assumption simplifies the annotation of multi-token entities to the level of POS annotation, where words are the unit of annotation. However, the risk of this assumption is that only part of the syntactic structure will be captured in the few cases where more complex noun phrases are used to express the concepts.

This assumption can be tested by comparing the number of potentially fragmented entities to the potential disagreement of annotators (which is the best available annotator error measure). The former can be approximated using the number of occurrences of

prepositions. As prepositions occur relatively rarely in the notes, surveying their usage is a feasible task. Instances of the prepositions "of" (253), "in" (244), "with" (192), "on" (183), and "from" (78) occurring outside the initial Read code terms representation were examined for potential use in concept-related complex noun phrases, e.g. *shortness of breath* (note that although frequently used in the GP notes this concept is predominantly expressed in abbreviated forms such as *sob*). Very few were found to be used in such context — 9, 10, 1, 7, and 0 respectively. Thus assuming that all concepts are base NPs carries a smaller risk of error than the human annotator agreement error margin for the Penn Treebank (Manning, 2011) – one of the best resources in terms of annotation quality.

5.1.2 *Guidelines Design & Annotator Training*

As the concept annotation task was formulated as classification of base noun phrases, the design of the guidelines was much simpler compared to the guidelines discussed in Chapter 3. It was no longer necessary to train the annotators to recognise target concepts in text. Instead, their task was limited to considering syntactic chunks in context, and allocating them to one of the classes defined by the guidelines.

Three classes of medical concepts were defined in the guidelines: *symptoms*, *disease names*, and *drug names*. Diseases are the names assigned to concepts of illness with a known cause, i.e. there is a known explanation for an affliction. Symptoms on the other hand, are the manifestations of diseases, but they may not always point to a single disease. Additionally, symptoms are often described as what the patient experiences, rather than what the doctor observes (King, 1982). The doctor's observations associated with a disease are called signs for that disease. For example, fever or dizziness are common symptoms reported by patients, but they are signs for a large number of diseases, e.g. flu, pharyngitis, and cancer of the larynx. A group of symptoms is referred to as a *syndrome*, however the term is used predominantly in the absence of a diagnosis. For example, the term *acquired immune deficiency syndrome* (AIDS) was used before the medical explanation for the symptoms was discovered — the *human immunodeficiency virus* (HIV) *infection*. The definitions of the three classes were made so that drug names refer only to names of drugs, disease names refer only to names of diseases, and symptoms refer to any of the following: symptoms, syndromes, and signs. The reason three slightly different concepts are combined

under one class is the subtle differences between them that may not always be apparent in the text. On the other hand all three have the same characteristic of being manifestations of a disease. Martin et al. (2014) take the same approach (not mentioning syndromes), giving as a reason the fact that both symptoms and signs are taken into consideration when a diagnosis is made.

The next step in the process was the selection of annotators. The logical choice was again medical experts, as the task depended only on medical expertise. There was a short process of guideline development and annotator training as the assumption was that the annotators would have a much better understanding of the concept classes and how to differentiate between them, than the author of the thesis. Therefore the guidelines aimed to provide a clear outline of the task, and the technical information needed for its execution, while keeping the class definitions as short and clear as possible (see Appendix B).

The guidelines arranged the adjudication of annotation disagreement in a different way to what was previously used for the Harvey Corpus. The task was delegated to the same annotators who needed to decide upon one class together, referring to medical dictionaries for term definitions. The reason there was no need for a third expert is that the only element where personal bias was likely to occur was co-morbidities, i.e. when a disease is also the sign for another disease. For example, *retinopathy* could be a sign of *diabetes*, but it could also be caused by *arterial hypertension*. In these cases it could be very difficult to decide which use the author intended, so the guidelines instruct that the *disease* rather than *symptom* class should be used in all such cases.

After the guidelines were completed, they were used to train two medical students as annotators in a two-hour training session. The training was followed by independent test annotation of a small data sample (not part of the Harvey Corpus). As in-text boundaries were no longer part of the annotation process, inter-annotator agreement had to be calculated using a different metric. Krippendorff's κ with equal label weights was the most suitable, as the annotation task was essentially multi-label classification. The measured agreement during training was 83%. A significant portion of the disagreement was caused by annotators failing to annotate concepts rather than intentionally not doing so. This can be explained by the fairly passive way the annotation platform works. The annotators are the active side browsing through text, and identifying concepts. However, that did not need to be the mode of working, as the number of annotations to be classified was

pre-established. Therefore, an extra non-concept annotation type was added to the guidelines in order to force the annotators to make a conscious decision for each annotation. In addition, they were given individual feedback on their trial work, and introduced to the new non-concept class. Given the high agreement rate and the improvements to the annotation process, it was decided that they were ready to start annotating the corpus.

5.1.3 *Analysis*

The corpus annotation process was much shorter compared to the process described in Chapter 3. The annotators achieved 89% κ coefficient, which is 6 percentage points better than the agreement from the training session. The final annotations were produced through a consensus round during which the annotators discussed their disagreement and reached a decision together. Table 5.1 shows a confusion matrix of the disagreement cases, showing that most disagreement is accumulated while differentiating between diseases and symptoms. In second place comes disagreement involving no annotation, i.e. the NPN category, which has two potential causes: disagreement on relevance and annotation error. When discussed with the annotators after their consensus round, it was established that it was a mix of both in roughly equal measures. It is curious to note that agreement on the NPDRG category appears to be quite high between the relevant categories while some disagreement (possibly due to human error) exists while deciding the relevance/existence of drug annotations.

	NPD	**NPDRG**	**NPN**	**NPS**
NPD	0	1	36	90
NPDRG	0	0	17	3
NPN	19	22	0	47
NPS	118	0	29	0

Table 5.1: A confusion matrix of the disagreement between the two annotators measured on the whole corpus annotation. Note that this matrix shows *disagreement* rather than the more usual *agreement*.

The relevant annotations included 1,169 taggings of symptoms, 482 of disease names, and 556 of drug names. However, only two thirds of these occurrences had unique string

representations. Around 85% of them occurred only once, while the remaining 15% accounted for 36% of all annotations.

Category	#	**Types**	**Singletons**	**Top Types**
NPS	1,169	897	782	pain (39), cough (17), wound (11), abdominal pain (11), diarrhoea (10), constipation (9)
NPD	482	356	303	malignant neoplasm (27), ca (12), mi (6), congestive heart failure (5), carcinoma (5), chest infection (4)
NPDRG	556	399	327	zoladex (22), paracetamol (11), casodex (8), frusemide (6), cyproterone (6), lactulose (6), ibuprofen (5)
NPN	5298	2767	1071	patient (105), he (85), telephone encounter (51), prostate (49), chest (43), hospital (32)
All	7,505	4,419	2,483	-

Table 5.2: Annotation statistics for the extended Harvey Corpus. Types: unique string representations of concepts. Singletons: types occurring only once.

The top entries in Table 5.2 show that the frequency distribution of annotation types (the unique string representations) is relatively flat with the exception of the two most frequent. These characteristics of the data suggested that concept recognition models may have difficulties if they relied solely on word-based features. This is further supported by the low average word type frequency of 2.26 inside the annotations — 2.68 for symptoms, 2.37 for diseases, and 1.73 for drug names.

5.2 TRADITIONAL CONCEPT RECOGNITION

The most common approach to named entity recognition encodes the target information as BIO token-based annotation to allow the use of sequence classifiers. The technique was previously demonstrated in the extrinsic evaluation of the initial semantic annotation of the Harvey Corpus in Chapter 3. This section reports a series of experiments set up in a very similar way to the CRFSuite optimisation experiments in Chapter 4. Inner cross-validation (Azzalini and Scarpa, 2012) with the same data splitting was used for model development, while an extra layer of semantic annotation was added to the dataset. Section 5.2.1 describes the optimisation of the feature types described in Chapter 4 for the purposes

of the current task as well as the testing of some new features. In contrast to the previous chapter where the source of POS annotation was a factor in the optimisation due to lack of a gold standard, the process here uses only the POS annotation selected as most favourable for the chunking models. This choice is important, because it ensures that the model development is optimal for real world applications, where models use features derived from dynamically generated annotation. Section 5.2.2 describes a comparison between the performance of optimised models using (static) gold standard annotation, and models using dynamically generated annotation. The difference between the two is crucial in the context of real-life applications where error propagation needs to be considered. The section also compares the strengths and weaknesses of the models in both contexts.

5.2.1 *Feature Set Optimisation*

The set of features explored in the previous chapter is underpinned by general properties of the processed language rather than characteristics specifically beneficial for chunking. Therefore, there is no apparent reason why those features should not be useful to concept recognition models. However, the optimal context windows for each feature type may differ from task to task. Adopting the best feature set configuration from Chapter 4 is unlikely to be the most optimal solution for concept recognition, but it can be used as a reasonable baseline for the optimisation. Using the feature set directly yields 54.79 f-score without using chunking annotation. Adding chunking features (containing type and border information) with the standard narrow context window increased the model performance to 63.98.

After a baseline was established, the optimisation process was carried out in three stages of cumulative improvement. First, the contribution of all features used for chunking models were evaluated using a narrow context window, in order to build a vector of feature types. Then their optimal context windows were determined through greedy group optimisation, and finally, Bayesian optimisation was used to tune the CRF hyperparameters.

Table 5.3 lists the feature types with positive contribution to the concept recognition model, as well as the performance resulting from their consecutive inclusion in the model feature vector. Compared to the optimal chunking feature vector (see Table C.2), the list of features is quite different. It seems that medical affixes are not as useful here as they

Growing Feature Set Models								
word	+	+	+	+	+	+	+	+
POS		+	+	+	+	+	+	+
chunk			+	+	+	+	+	+
Ney-Essen				+	+	+	+	+
Brown					+	+	+	+
canonical						+	+	+
suffix							+	+
auto affix								+
f-score	45.74	47.91	56.81	60.72	64.25	64.61	68.32	69.46

Table 5.3: Features with positive contributions to concept recognition models. Columns represent models, pluses indicate presence of feature. The bottom row shows the f-score achieved by the model.

were to chunking, as opposed to automatically generated affixes, which contribute more than 1.5 percentage points to this task (note that these are in fact two feature types — automatically generated suffixes and prefixes each contributing roughly the same). The generic suffixes also show better performance when used together rather than separately based on part-of-speech. Another difference from the chunking vectors is the positive contribution from both Ney-Essen and Brown vectors — the use of both word cluster features did not lead to better performance in chunking even though they contributed significantly when used separately. Finally, none of the n-gram features were used because they all had a negative effect on the model performance.

One aspect of the feature vectors that needed additional attention was the sources of the cluster features. The best performing clusters were used for the initial experiment, but it was not clear whether they were the most optimal choice. Two series of experiments were conducted to determined the best combination of Ney-Essen and Brown clusters. The first series tested the best performing Brown clusters using the feature vector scheme in Table 5.3 and the 250 Ney-Essen clusters trained on the GPRD data. The results of the experiment showed that the 250 GPRD Brown clusters performed better than the rest although not all of the differences were statistically significant. The second series of experiments also found that the initially selected cluster set was the most suitable for the task, although the difference from the Stanford clusters was not significant (69.23, p=0.616).

Once the choice of feature types was finalised their context windows were optimised using the group optimisation technique discussed in Chapter 4. The nine feature types were grouped in threes: 1. *word*, *POS*, *chunk*; 2. *canonical*, *Ney-Essen*, *Brown*; 3. *suffix*, *automatic prefix*, *automatic suffix*. The context window optimisation process yielded a very slight improvement in performance reaching 69.80 (p=0.41). Finally, Bayesian optimisation was used for tuning the two CRF hyperparameters, which increased the performance more significantly, reaching 70.68.

5.2.2 *Performance Analysis*

The f-score of the concept recognition model measured in the final stage of the inner cross-validation evaluation was 65.93. It is difficult to compare this performance to that achieved by other researchers, not only because this is the first model used on this corpus, but also because related tasks focus on different concepts or different data. Nonetheless, with some exceptions (Pyysalo et al., 2013; Stenetorp et al., 2012b), NER-like tasks typically score in the high 80s or even low 90s (Tjong Kim Sang and De Meulder, 2003; Y. Guo and R. Gaizauskas and I. Roberts and G. Demetriou and M. Hepple, 2006). It therefore seems that the model performance is relatively low, but there is also the question of the task complexity and the reasons behind the errors of the model.

The model achieved 69.59 precision and 62.63 recall; immediately noticeable is the comparatively low recall and the large precision-recall gap. Such a large gap may be indicative of a BIO-task optimisation problem noted by Manning (2006). In the case of NER, and by extension other tasks using BIO annotation, optimising for F_1-score often leads to a conservative model with lower recall. The reason for this bias is rooted in the definition of a correctly recognised concept. In the case of simple classification, data points fall into one of four classes — *true positives* (tp), *false positives* (fp), *true negatives* (tn), and *false negatives* (fn) — while tasks using BIO annotation have a complex definition of a data point, which causes errors in border recognition and label classification or both. Manning describes an experiment where three new categories of errors are introduced: *label error* (when the boundaries fully overlap but the labels are different), *boundaries error* (when the labels are the same but the boundaries only partially overlap) and *label and boundary error* (when the labels are different and the boundaries partially overlap).

He shows that in a classic NER experiment the majority of errors fall into one of those categories as opposed to the classic four. His argument is that f-score optimisation punishes these categories harder than the rest thus favouring more conservative (passive) models.

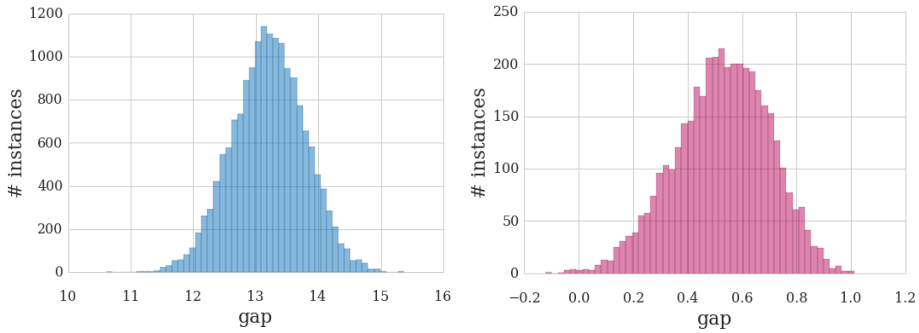
A somewhat involved but cleaner explanation to the problem can be derived from the mechanics of calculating f-score for BIO tagging. The f-score metric makes intractable evaluation tractable. In the evaluation of information retrieval systems (e.g. TREC (Voorhees and Hersh, 2012)), documents are only considered if they have been retrieved by one of the systems (i.e. marked as positive). For retrieval problems this is absolutely fine but for BIO annotation, there is a small problem: retrieving some documents makes the retrieval of others impossible. For example, consider the symptom *abdominal pain* and the following wrongly guessed annotation:

Guess: `abdominal/O pain/B-NPS`

Gold: `abdominal/B-NPS pain/I-NPS`

The system has produced a partial match and missed the real one. So there is a new document in the universe that is a *false positive*, which is normal, but at the same time it also becomes impossible for the model to produce the right document. So essentially, with one action the model makes two decisions: 1. creates a new document which is a *false positive*, and 2. gives up on finding the correct one which is a *false negative*. Considering this imbalance, it is possible that a model with an unusually large precision-recall gap was selected during the model development. The rest of this subsection investigates that possibility by analysing the precision-recall gaps of the development models across the relevant experiments.

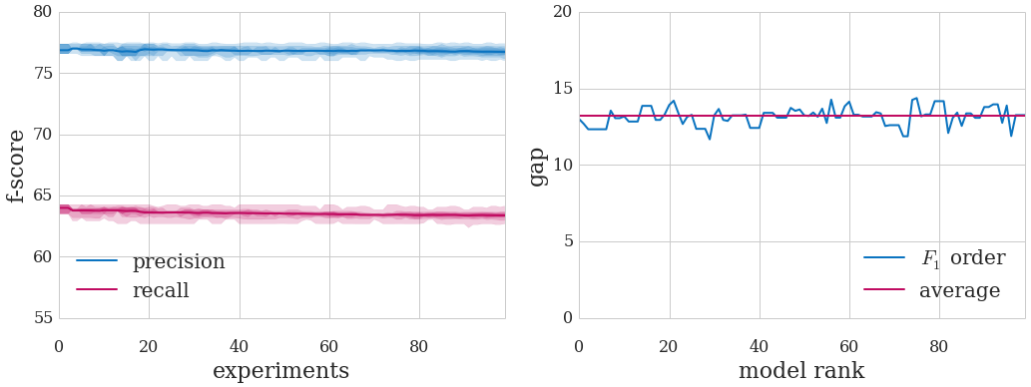
The simplest analysis step (apart from checking if the first few models have similar precision-recall gaps) is to look for outliers as well as the distribution of the gap size across all development experiments involving optimisation of BIO annotation models. Figure 5.1 shows a histogram of those sizes which shows that the gaps in the concept recognition model during the development process are even bigger than in the final evaluation. They are also much larger than the precision-recall gaps of the chunking development model (again a BIO model), which are mostly less than 1 and in some cases even negative (meaning recall is greater than precision). Due to the difference between the performance in development and the final evaluation, it is not possible to determine only from the gap



(a) concepts (b) chunks

Figure 5.1: A counts histogram of the precision-recall gap in concept recognition (a) and chunking (b). The estimates are based on data from the greedy group optimisation experiments.

distribution whether the optimisation process was misled, although it is clear that recall is generally much lower than precision across all development results.



(a) (b)

Figure 5.2: (a): Average precision and recall values of the top 100 development models across β values between 0.5 and 2. (b): precision-recall gap for the top 100 development models compared to the average across all models.

A more detailed analysis of the precision and recall results in the development population indicates that they are both relatively stable. Figure 5.2b shows little variation for both precision and recall for the top 100 models (based on f-score measured during the development process) for β values ranging between 1 and 2. Both Figure 5.2b and 5.2a reveal a fairly consistent precision-recall gap, while the second also indicates that the gap for the best performing model is lower than the average for all development models. Perhaps the only place that presents some evidence of a recall bias is the comparison of projections of top f-score for different values of β shown in Figure 5.3. The top model for β=1 coincides with the one for β=2, and performs slightly worse than the top model for

β=3. However, the absolute difference in performance is not statistically significant for any plausible α threshold (p=0.109 using a Wilcoxon test), so it is fair to conclude that the recall optimisation bias was not decisive for model selection.

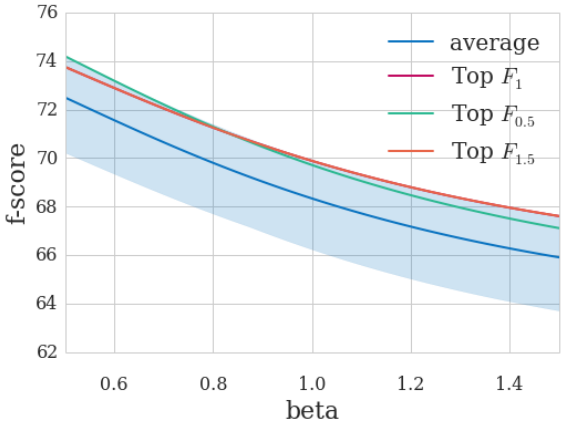


Figure 5.3: Average f-score distribution of development results across different β values, compared to the distribution of the top f-score for β=0.5,1,2. The shaded area indicates variation boundaries.

Since recall levels across the development experiments are consistent and there are no specific reasons for a significant bias, it is possible that the low levels are caused by the high proportion of words occurring only once, as mentioned in Section 5.1.3. This is likely to be the cause of border errors as well as of standard recall errors (missing concepts). Additionally, border errors impair both precision and recall, which is consistent with the observed development results seen in Figure 5.2b. However, Table 5.4 indicates that the proportion of tokens erroneously classified outside a concept is consistently greater than that of border and label errors. This is also confirmed at the concept level, where 42 out of 190 concepts were not detected at all, compared to 14 label errors and 5 border errors. This behaviour suggests that the classifier has not acquired enough information to recognise concepts containing unknown words. Additionally, there were fewer (24) spurious annotations, i.e. false positives in places without any gold standard counterpart, than missing annotations, which indicates that the classifier is better at recognising what is *not* a concept. This analysis, in conjunction with the low repetition rate of concept words in the corpus, indicates that more training data is needed to strengthen the word features in the classification model.

	B-D	B-DR	B-S	I-D	I-DR	I-S	O	All
B-D	-	-	6.38	2.13	-	-	34.04	42.55
B-DR	-	-	-	-	-	-	24.32	24.32
B-S	10.38	-	-	-	-	-	18.87	29.25
I-D	2.33	-	-	-	-	9.3	25.58	37.21
I-DR	-	11.11	-	-	-	-	44.44	55.55
I-S	-	-	-	11.63	-	-	15.12	26.75
O	0.22	0.13	0.71	0.49	0.09	0.94	-	2.58

Table 5.4: Token-level errors analysis by category as percentage of all annotation in that category. Rows are gold standard categories, columns are inferred categories. B: begin; I: inside; O: outside; D: disease; DR: drug; S: symptom.

5.3 AN ALTERNATIVE APPROACH: DIVIDE & CONQUER

Most machine learning approaches to NER-like tasks use methods based on BIO-annotated tokens, which combines recognising the borders of an entity, and determining its class. The task is rather similar to chunking and POS tagging in terms of classification mechanics, but it is marginally more difficult to train a good classifier for it. Typically, the targeted entities are within some form of a syntactic constituent such as a NP, but their exact boundaries do not necessarily coincide. This relationship between entities and syntactic structure is the reason syntactic features play a key role in building a successful statistical model for an NER-like task, as was demonstrated in Section 5.2. However, the parameters of the task discussed in this chapter allow for a significant simplification due to the terse language used in primary care notes. The targeted medical concepts happen to be mostly base NP chunks, so the borders of all potential concepts could be inferred automatically from the chunk annotation by making the assumption that they are all also NP chunks. Following this approach, referred to hereafter as the Divide & Conquer (DC) approach, the classic NER-like task could be simplified to a multi-class document classification where NP chunks are treated as documents. Even though the assumption that NPs can be used as boundaries for entities is new, the idea of dividing up an entity recognition task has been suggested before (Carreras et al., 2002; Wang and Patrick, 2009). Wang and Patrick (2009) describe a similar approach to NER in clinical notes as cascaded classifiers. They use CRFs to identify the boundaries of entities, while leaving their labelling to an ensemble of a SVM classifier and a MaxEnt classifier. They report 88.12 f-score in recognising eight

types of entities in 311 admission summaries from patients admitted to a hospital intensive care unit.

The fundamental difference between the previous and the proposed approaches, namely the explicit knowledge that concepts are always NP chunks, suggests that their results should not be comparable. However, this can be determined empirically by looking at the annotations made by the model using the previous approach. Only one out of 674 concept annotations was created at a place where a chunk did not begin. This is solid evidence that the model has learned what is available to a DC model by design, and so comparing their performance should be possible.

Even though the data and the types of features available for extraction remain practically the same as what was described in the previous section, the use of a document classification approach for the concept recognition task required the whole model development process to be revisited. Support vector machine classifiers are commonly used for such tasks, and are typically compared to a Naïve Bayes (NB) baseline. However, the choices of SVM kernel and multi-class classification strategy play an important role in the behaviour of the trained models. Feature engineering is also approached differently in classic document classification compared to structured classification (e.g. by CRFs). However, the small size of the document collection in this model development also allowed the use of both positional and bag-of-words features.

The experiments were set up considering the text within all NP chunks as documents and their associated concept annotation as labels. The non-concept annotation introduced for the convenience of annotators was assumed to be the negative label, meaning that correctly classified non-concepts were not counted as true positives, but as true negatives. This was necessary in order to ensure fair assessment of the classifier performance on the target concepts. The model development process was carried out using inner cross-validation scheme in the same way as for the traditional approach. It should be noted that the f-score calculation had to treat the non-concept class as a negative label as opposed to a class of equal importance in order for the evaluation to be fair (see Algorithm C.1 for a Python implementation).

5.3.1 *Surveying Classifiers*

There is a wide variety of classifiers that are able to deal with the document classification. Multi-class classifiers such as multinomial Naïve Bayes, k-nearest neighbours (kNN), and decision trees are a natural fit for the task, while binary classifiers such as linear kernel SVMs can be adapted to multi-class classification using one-vs-rest or one-vs-one strategies. Table 5.5 compares the performance of these classifiers on the concept classification task using a simple set of positional word and POS tag features, and a bag-of-words (BoW) set of the same feature types (with the feature-value representations in Example 5.1). The positional features link the extracted information with a position in the document, while the BoW approach disregards word positions and concentrates on occurrence counts in the documents.

{PAIN: *1*, SEVERE: *1*}	{WORD1: *severe*, WORD2: *pain*}
(a) Bag of words	(b) Positional

Example 5.1: Feature extraction approaches giving feature-value pairs for *severe pain*.

SVMs achieve the highest results using both feature sets, but while their advantage is statistically significant for positional features, multinomial Naïve Bayes achieves a comparable performance using the bag-of-words approach. Additionally, the reported results were achieved after some initial investigation of the classifier parameters using Bayesian Optimisation. The process produced some interesting findings, such as the influence of SVM kernels as shown in Table 5.6.

	Positional	**Bag of Words**
Naïve Bayes	50.42	73.21
kNN	45.51	42.50
Decision Trees	57.87	70.40
SVMs	**66.91**	**73.34**

Table 5.5: F-score of classifiers on the concept classification task using a positional and a bag-of-words style feature set.

The low performance of kNN is most likely caused by the small size of the training set, and the imbalanced number of non-concepts compare to the rest of the classes. It has a particularly low recall (34.21) indicating that the non-concept class (which is the negative class) is favoured by the classifier, which means that it outvotes the rest even in cases where it should not.

Kernel	**Strategy**	**Positional**	**BoW**
Linear	one-vs-rest	**66.91**	73.03
Linear	one-vs-one	66.19	73.28
Polynomial	one-vs-one	66.15	**73.34**
RBF	one-vs-one	65.21	56.39
Sigmoid	one-vs-one	30.36	48.08
Crammer and Singer (2002)	multi-class	65.57	73.19

Table 5.6: Performance comparison between different SVM kernels and multi-class strategies.

Even though it was interesting to investigate the impact of different SVM kernels, multi-class strategies, and hyperparameters, any conclusions drawn at this point may not have held at a later stage of development using a richer feature model. To a certain extent this was also true for the choice of a classifier, so it was decided that SVMs (which were the top performing classifiers from this round of experiments) should be used for model development, after which another round of experiments was needed to give more context to the comparison.

5.3.2 *Feature Engineering: Bag of Words*

The main part of the model development process needed to explore four directions of variations regarding the SVM models: both positional and BoW feature extraction approaches needed to be considered; a range of feature types derived from their CRF counterparts had to be tested; and finally the choice of kernels and their specific parameters and hyperparameters. To explore these possible model variations two groups of experiments were set up — one for each feature extraction approach. Each group explored separately the effects of newly crafted features, the use of kernels, and hyperparameters.

The features found helpful during the CRF-based model development in the previous section of this chapter were also considered in the context of BoW features for document classification. Only one feature type was left out of the experiments because it was expected to be of little use due to the new structure of the task — the chunk annotation. The rest of the feature types discussed in Section 5.2 were used in the experiments, but none of them achieved any improvement over the initial model. Some of them (POS bigrams, suffixes, prefixes) had no or very little effect on the model performance, while others degraded it significantly (word embeddings, Brown and Ney-Essen clusters).

Another attempt at improving the performance looked at adding some context to the document, which meant including the words preceding and following the NP chunk in the document. This approach had both potential advantages and disadvantages. If the context meant that a verb can be recognised around the document that could be useful, as, for example, in the case of *prescribe_MV fybogel_NP*. But context could also be misleading if two NP chunks are next to each other as in *no coughing high fever*. Experiments showed that introducing context tokens led to a severe drop in performance that was proportional to their number.

However, the model was improved by the least expected feature type — chunk annotation. The only information that it could contribute in its current BIO state was to differentiate between the first word in a document and the rest. The new model received a performance boost of a little more than half a percent f-score reaching 73.92 which could be considered significant at an undemanding threshold (p=0.028).

Even though there was very little gain from the model development process, the development results were still higher than the f-score achieved by the CRF model, which could be considered a good baseline for the model.

5.3.3 *Feature Engineering: Positional Features*

The positive effect of the chunking annotation features for the BoW model indicated that word position is important, which gave further support for investigating positional feature extraction. All feature types described above were also tested using the positional feature extraction approach.

Adding chunk annotation features to the baseline word and POS annotation model resulted in a slight statistically insignificant drop in model performance (66.83). This is not too surprising given that the much more informative POS annotation increases the performance of a purely word feature driven model from 66.30 to 66.91, a borderline significant difference (p=0.013).

The context features discussed above generally had a very small mostly negative effect on the model performance, except for the word immediately preceding the document which improves the model performance to 68.42 (all results available in Table D.5). Note that the reported results include both word and form features from the context position, each of them contributing a roughly equal share. An additional experiment was conducted to test if the chunking annotation may be useful under these conditions, given that it will highlight the border between a concept and non-concept, but the results showed an insignificant dip in the performance to 68.37 rather than an improvement.

The addition of each word representation feature type increased the model performance more than any other discussed so far for this task. Additionally, their influence in absolute percentage points was also more than in any other experiment described in this thesis (results from all word representation experiments are available in Tables D.6, D.7, & D.2). Additionally, the results shown in Table 5.7 indicate that combining two or three of the feature sets also leads to improvements, in contrast to what was observed in the chunking experiments in Chapter 4, where combining them did not lead to any improvement. It is worth noting the consistently good performance of the GPRD cluster features. Finally, it is interesting that all word embeddings lead to some positive effect, but only the features based on the Dhillon et al. (2015) embeddings showed a statistically significant improvement.

Word n-grams and fixed-size substring affixes also proved to have a positive effect when used with positional feature extraction. Both word bigrams and trigrams increased the f-score by more than one percentage point to 75.83 and 75.54 respectively, but their combined use did not improve upon bigrams alone. Similarly, prefix and suffix features achieved almost the same improvement on their own — 78.26 at size three and 78.30 at size two respectively — but again combining them did not reap any improvement over suffixes alone (78.28).

	source	size	f-score	precision	recall
Embeddings	Dhillon	200	69.78	79.70	62.37
Brown	GPRD	512	72.67	79.39	67.00
Ney-Essen	GPRD	512	71.58	77.85	66.40
B+NE	-	-	73.77	77.46	70.68
All	-	-	74.57	78.23	71.54
Baseline	-	-	68.43	83.67	58.15

Table 5.7: Comparing positional DC performance impact of word representation features, separately and together.

(a) {WORD1: *severe*, WORD2: *pain*}

(b) {WORD1: *possible*, WORD2: *severe*, WORD3: *pain*}

Example 5.2: Left alignment of positional features in the documents (a) *severe pain* and (b) *possible severe pain*.

Additionally, the choice of left or right alignment of positional features was reviewed, since NPs usually have their head word in the rightmost position. If left alignment is used as illustrated in Example 5.2 then the common features WORD1 and WORD2 will have different values. If feature extraction is aligned to the right then the common features will have matching values. Using right alignment significantly increased the f-score to 80.00.

There were a few further model improvement avenues that seemed worth investigating, but did not help. Character n-gram features, which are essentially a more generic version of the affix features, consistently decreased performance (77.01) so they were left out. Count-based word features were also explored, because of their importance for the BoW model performance, but eventually they ended up slightly decreasing the performance of the positional model.

5.3.4 *Feature Selection*

The purpose of feature selection is to sieve out the features with high entropy, leaving only the "useful" ones in the set. In fact, considering that a feature set already existed before looking into the DC approach, the process described in the previous two sections is similar to forward stepwise selection (Caruana and Freitag, 1994). A more principled approach uses a statistic to weed out features that fail to discriminate between classes by

picking a subset of them that score over some threshold. Two common statistics used for this purpose are the ANOVA F-test value (Markowski and Markowski, 1990), and the χ^2 statistic.

ST	FS	classifier	10%	20%	50%	90%	crafted
χ^2	P	SVM_{RBF}	79.03	80.17	62.92	42.45	45.32
F-test	P	SVM_{RBF}	78.89	80.05	62.92	42.45	45.32
χ^2	P	NB	78.25	79.47	80.10	71.62	67.95
F-test	P	NB	78.22	79.47	80.10	71.62	67.95
χ^2	P	SVM_{Linear}	79.94	78.04	76.56	79.45	80.00
F-test	P	SVM_{Linear}	80.02	78.06	76.56	79.45	80.00
χ^2	BoW	SVM_{Poly}	69.96	69.54	68.89	69.15	73.91
F-test	BoW	SVM_{Poly}	69.24	68.78	66.09	66.24	73.91

Table 5.8: Comparison of models using feature selection to a model using a manually crafted feature set. The crafted column shows the classifier performance using the best crafted feature set. ST: statistic, FS: feature set type, P: positional feature set, BoW: bag-of-words features, RBF: radial basis function

Table 5.8 lists results from eight experiments which tested feature selection using F-test and χ^2 on the full bag-of-words feature set and the full positional features set (see Table D.9 for all experiments). In many cases, especially for models using BoW features, the feature selection process led to a significant drop in performance, but in other cases the process led to significant improvement. Even though the highest result was achieved by the RBF kernel SVM classifier using the top 20% of the features selected using χ^2, each of the shown models with position-based features achieved comparable results, meaning without statistically significant difference. Given the close scores the choice of a model needed to be motivated by different factors.

classifier	ST	FT	f-score	σ	precision	recall
SVM_{Linear}	-	crafted	80.00	1.82	84.16	76.38
SVM_{Linear}	F-test	10%	80.02	1.63	88.83	73.01
SVM_{RBF}	χ^2	20%	80.17	2.18	79.29	81.07
Naïve Bayes	F-test	50%	80.10	2.25	77.31	83.09

Table 5.9: Precision-recall gap comparison in top scoring models. ST: statistic, FS: feature set type.

The model using a linear SVM kernel was chosen as the safest model as it is least sensitive to feature selection across different thresholds, which indicates that it is less prone to overfitting. The models using feature selection and a crafted feature set achieve virtually the same result, in contrast to the Naïve Bayes and the RBF kernel SVM models. The latter two also have a higher f-score standard deviation (see Table 5.9). The small precision-recall gap of opposite polarity is the only advantage that other models have over those using a linear kernel SVM classifier. However, that may also be considered a drawback if the overall goal is high precision.

5.3.5 *Performance Analysis*

The model using linear kernel SVM with top 10% F-test feature selection yielded surprisingly high results on the final test set. Its f-score reached 83.76, which is significantly higher than the development levels, but more impressively the precision-recall gap had shrunk down to 1.43 percentage points (84.48 precision and 83.05 recall).

	diseases	**drugs**	**symptoms**	**none**
diseases	-/-	-/-	5.33/1.75	12.00/14.04
drugs	-/-	-/-	2.67/-	13.33/12.28
symptoms	-/15.79	1.33/1.75	-/-	42.67/26.23
none	1.33/10.53	5.33/7.02	16.00/10.53	-/-

Table 5.10: Label error rate comparison between the development set (left; mean values) and the test set (right) as a proportion of all correct occurrences of this label. Rows signify correct labels, columns labels assigned by the classifier.

Comparing the rate at which true labels are mistaken for other labels is a good way to identify what has improved between development and the final test. Table 5.10 shows that the Type II errors for the *symptom* class (rightmost column) have decreased by more than 15 percentage points, while Type I errors have increased overall. These observations suggest that the model behaves in a less conservative way than on the development data.

Even though the performance of many of the models using the DC concept recognition approach reported so far are much higher than the models using the traditional approach, a fair comparison of the real-life applicability of the two approaches requires assessment of their vulnerability to substandard chunking annotation. Both approaches rely on accurate chunking. On one hand, a chunking border mistake does not necessarily entail a concept recognition mistake for the traditional model, but it makes it very likely. The model has a chance of recognising the concept with the correct borders (depending on the other features), but typically it would either not recognise the concept or make a border error. In contrast, the DC model has no possibility of making a correct decision in this situation. On the other hand, a chunking label mistake always entails a non-concept recognition for the DC model, which is correct in most cases, since most NPs are non-concepts. A traditional approach model is less likely to recognise a concept without chunking annotation, but it is not impossible, as shown in earlier experiments without chunking features (see Table 5.3). So following this rationale the traditional approach should be at an advantage if substandard, *dynamically* generated chunking annotation is used in the final validation set as opposed to the *static* gold standard.

The same data splitting was used for concept recognition as for the chunking experiments, so the inferred chunking annotation for the validation set in Chapter 4 could be used for the dynamic evaluation of concept recognition. This validation set was used to re-evaluate the two best performing models of each approach, and it was found that although the performance of both methods drops severely, the Divide & Conquer approach still retains a large advantage over the traditional approach. The latter achieved 57.85 f-score (60.69 precision, 55.26 recall), while the former achieved 70.67 f-score (70.32 precision, 71.03 recall). These results indicate that the DC fares better than the traditional approach even in unfavourable circumstances. It should be noted, however, that the percentage points lost due to dynamic chunking annotation is significantly greater for the DC approach in both absolute and relative terms.

5.5 CHAPTER SUMMARY

This chapter presented the design and development of approaches to clinical concept recognition, covering symptoms, diseases, and drug names. An extra layer of annotation was added to the Harvey Corpus to enable the training of such recognition models. Two annotators were trained using a set of guidelines based on the assumption that the targeted concepts are always noun phrases. The corpus was annotated independently by each annotator, with inter-annotator agreement reaching 89% Krippendorff's α. A novel semantic entity recognition method was proposed, which approached the entity identification and classification problems while utilising the terse NP-heavy characteristics of primary care data. A comparison between this approach and the traditionally used sequence taggers showed that the former achieves better f-score performance both when using gold standard and dynamically generated chunking annotation.

The f-score of a sequence tagging model developed using inner cross-validation yielded 69.59, which was disappointing. However, the model development proved word representation features, including word embeddings, to be much more important than they were for the chunking models discussed in the previous chapter. Due to a large difference between the precision and recall of the model, it was suspected that the optimisation process had led to the selection of a conservative model favouring low recall in order to boost f-score. An analysis of the performance of all development models showed that the size of the precision-recall gap is not unusual, and that the selected model would have been the same even if a recall-favouring f-score was used (β=2). Additionally, an error analysis of the results suggested that the classifier was much more likely to fail to recognise concepts rather than their classification or borders, which can be indicative of insufficient training data.

The alternative approach proposed in this chapter (called Divide & Conquer) took advantage of the NP-heavy primary care notes to eliminate the border recognition part of the task. It assumes that all of the target concepts must be noun chunks, as symptoms, diseases, drug names are undoubtedly centred around nouns. Thus the concept recognition problem becomes an NP chunk classification problem with four classes, one of which is a non-concept (negative) class. The new approach achieved results far better than the traditional approach during the development stage of the inner cross-validation, and achieved

83.76 f-score on the validation set. The model development process tested the performance of typical document classification algorithms, while following two feature engineering approaches, one using standard document classification bag-of-words features, and one using positional features designed in a similar way to sequence modelling tasks. Additionally, hand-crafting a feature set was compared to feature selection techniques based on χ^2 and ANOVA F-test values. The positional features proved to give overall better performance than BoW, while the hand-crafted feature set used with a linear kernel SVM was among the top performing models, along with three others using feature selection and linear kernel SVM, RBF kernel SVM, and Naïve Bayes classification. The final model was chosen to be the 10% F-test feature selection model using linear kernel SVM. The decision was made based on the stability of linear kernels and the assumption that feature selection based on F-test value generalises better than the hand-crafted approach (although the development performance was virtually the same).

Perhaps the most important test described in this chapter was measuring how using substandard chunking annotation affects both methods. It was established that the Divide & Conquer approach still performs better both in absolute and relative terms, although its f-score dropped significantly to 70.67. This finding is particularly important as it gives an indication of the potential real life performance of the model presented in this chapter.

In conclusion, the chapter presented a statistical model for recognising symptoms, diseases, and drug names based on a new annotation layer of the Harvey Corpus. Although the model did not achieve results as high as recent entity recognition models for edited text, it still demonstrates that relatively reliable results can be achieved despite the non-standard nature of the language of primary care notes.

❧ ❋ ❧

6

DISCUSSION & FUTURE WORK

6.1 THESIS SUMMARY

This thesis presented the research efforts behind the development of a novel clinical concept recognition system that is able to handle the terse and non-canonical language of primary care electronic text notes. The system is the first of its kind suited for the language in the text part of UK primary care electronic medical records. It has to account for spelling and grammar errors, abundant terminology, terseness of expression, and numerous acronyms and abbreviations, which are difficult for any generic natural language processing system or model to deal with correctly. Additionally, the amount of available data resources was restricted due to the presence of sensitive information, which imposed further difficulties in the development process. The work described in the thesis addressed these challenges by developing new language resources and statistical models suited to primary care text processing. The overall strategy to achieving robust information extraction results involved selecting the most suitable existing part-of-speech (POS) model rather than develop it from scratch, and concentrating on creating language resources and models for syntactic chunking and concept recognition.

The first step towards the development of statistical models suited to a new type of text is creating an adequate language resource, i.e. an annotated text corpus. Chapter 3 described the first stages of the development of the Harvey Corpus of primary care text, in which a random sample of electronic medical records was selected for annotation. A set of guidelines were created through an iterative development and evaluation process involving a group of researchers and a pair of annotators with medical training. The selected records were annotated with syntactic chunk annotations specifically developed for

this type of text, and four types of semantic entities. The annotation process involved two annotators creating independent versions of the annotations, and an adjudicator reviewing and resolving the cases of disagreement. The inter-annotator agreement reached an average f-score of 85 for the chunking annotation and 71 for the semantic entity annotation. The corpus eventually included 850 electronic medical records constituting more than 25,000 tokens. Another annotation layer was added at a later stage when the corpus was prepared for the development of concept recognition models. The process was slightly simplified as it was assumed that medical concepts occur almost exclusively as noun phrases in the context of primary care text. Thus, the annotated noun phrases were upgraded with medical concept annotation, covering symptoms, diseases and drug names. The annotators achieved 89% inter-annotator agreement, measured using Krippendorff's α coefficient.

The Harvey Corpus enabled the development and evaluation of statistical models for processing primary care data, which led to a methodical process of model feature experimentation and model development for syntactic chunking and concept recognition, using different machine learning algorithms. Chapter 4 explored in depth the best approach to achieving good results in the POS tagging and chunking tasks. Existing models for both tasks were evaluated on the Harvey Corpus, yielding 80.69% POS accuracy and 46.41 f-score for chunking. Considerable compromises had to be made in order to perform this evaluation, so the results were treated as indicative of trends rather than absolute performance. Nevertheless, the rationale was to rely on available POS models, while concentrating on developing chunking models. The next step was optimising existing tools and methods for chunking using the Harvey data, which achieved chunking performance reaching 77.76 f-score. The experiments revealed that a CRF-based tool performed better than an SVM-based tool, and that although BEISO annotation achieved higher results in more of the experiments, there was no significant difference between any of them. A more flexible machine learning tool was developed based on the *CRFSuite* library, allowing the optimisation of a wider range of commonly used feature types including word representation clusters and vectors. The chapter also investigated the issue of multi-parameter optimisation, exploring Bayesian optimisation for feature set development, and comparing it to isolated grid search across hand-crafted groups of feature types. The development process produced a model that achieved 79.89 f-score on the final validation set.

Chunking and clinical concept annotations allowed the development of the final step of the concept recognition pipeline. The task was approached in both the traditional NER-like way, and using a new method exploiting the specific primary care language features to its advantage. The traditional approach followed roughly the same experimental setup as the chunking task with some modifications and new features adapted to the task. The results achieved using that method were much lower than results typically reported in clinical concept recognition studies, reaching only 65.93 f-score. The results had an unusually wide precision-recall gap with a skew towards precision, which seemed consistent with a F_1-score optimisation bias previously suggested by Manning (2006). The scores from the whole optimisation process were recalculated optimising for the whole spectrum between $F_{0.5}$ and $F_{1.5}$, but there was very little change in the configuration order, which showed that even if there is a bias due to optimising for F_1-score it was not what caused the gap.

The alternative approach, called Divide & Conquer, took advantage of the telegraphic style of the text which generally packs information, including medical concepts, in base noun phrases, i.e. NP chunks. Using that domain specific knowledge, the method assumed that all NP chunks are potential candidates for concepts, which transformed the recognition task into a text string (or document) classification task as borders no longer needed to be recognised. Traditionally, document classification tasks favour bag-of-words (BoW) features over positional features for a number of reasons, but in the case of phrase sized documents, considering word positions seemed important, so both types of feature engineering experiments were pursued. The validation scores of the positional feature model outperformed the BoW best model by more than fourteen percentage points, reaching 83.76 f-score compared to 69.59.

Finally, the two approaches were also tested in a real-life system setup, where both the POS and chunking annotations were generated automatically. Even though positional features dominated so decisively, it was interesting to see how dependent they would be on the quality of the chunking annotation and to compare that dependence to the bag-of-words model. The performance of both models decreased as would be expected, but surprisingly the loss on both sides was very close. The positional model achieved 70.67, while the BoW performance dropped to 57.85. The performance drops were very similar in absolute terms (under a percentage point apart), but the positional approach lost less from

the annotation in relative terms, which is slightly unexpected given its strong dependence on good chunking annotation.

The development of a clinical concept recognition system for primary care text along with the research that led to it has interesting implications for the field of clinical NLP, as well as epidemiology and other forms of clinical research. This work showed that despite the difficulties posed by the non-canonical language in primary care free text notes, automated analysis is still possible through adapting existing technology and exploiting the terseness of the language. Additionally, the prototype system presented here is a major step towards developing robust task-independent tools to aid medical researchers in exploring primary care data on a large scale.

6.2 MAIN FINDINGS

The main findings revealed over the course of the research conducted for this thesis are briefly described below.

During the development of the chunking guidelines, it was noted through the logs of the annotation platform that annotators perform qualitatively better when working for longer periods of time. Annotations produced during sessions shorter than thirty minutes were found to be of considerably lower quality compared to those produced during longer sessions. In relation to this finding, it was also noted that generally the quality of the first few records annotated during a session is lower than the average for the session. The annotation process was adjusted in order to account for these negative effects — annotators were assigned longer, supervised sessions, and were encouraged to review their work at the end of each ten-record annotation batch, as well as whenever necessary in general.

Word representation features were an important part of the experiment base of this thesis. Word cluster features were found to be particularly useful in the BIO-based tagging tasks (chunking and traditional concept recognition), as well as the document classification task (Divide & Conquer concept recognition). In contrast, the influence of word embedding features on the performance of the two BIO-based tagging tasks was generally found to be statistically insignificant. They did, however, contribute significantly to the models using the document classification approach. Additionally, the experimental findings were complementary to the findings of Stenetorp et al. (2012b), as the word representation

features based on in-domain (i.e. primary care) data performed on a par with those based on biomedical text and general domain text, in spite of smaller data set size.

Bayesian optimisation, an optimisation method that has recently grown popular among machine learning researchers, was successfully used to tune hyperparameters of NLP classifiers, but it was found to be better than greedy parameter group optimisation only in some cases of feature set optimisation. The experiments showed that Bayesian optimisation was several times faster when the optimised feature set had four or less tunable feature types, but it did not improve on the performance of greedy parameter group optimisation when tuning eight feature types — even after a very high number of repetitions. Even though the experiments suggest so, it is still uncertain if the poorer performance with more tuning parameters (i.e. feature set types) is due to the higher number of combinations. Another possible explanation is a greater or more unpredictable impact of adjusting feature set types compared to adjusting hyperparameters, which would make the task more difficult to model.

Syntactic chunks are usually good indicators of the borders of semantic entity annotations, but there are many exceptions. Therefore the general practice is to detect borders together with entity classes in a BIO annotation, while using chunk features. However, in this thesis it is assumed (based on an empirical analysis) that medical concepts are predominantly expressed as base noun phrases, which allowed the direct classification of predefined text units (documents) into medical concept (or non-concept) classes. Additionally, even though such a classification approach seems to have an unfair advantage over the traditional approach, in practice the advantage was negligible as the developed BIO tagging model made almost no border errors on the same task.

6.3 CONTRIBUTIONS TO NLP & CLINICAL RESEARCH

Health researchers rarely use NLP tools when carrying out studies on primary care text due to the sensitive information contained in the text, and the difficulty of processing this kind of language with tools and models trained on text from a different domain. The main contributions of this thesis are a clinical concept recognition system for UK primary care text, and the exploratory work through which the system was developed. Even though the system's accuracy did not reach that of similar techniques in other text

domains, it provided an indication that semantic information can be extracted from GP text notes. In essence this work should be considered as an initial step towards large-scale extraction of information from primary care text notes in clinical and other medical research. Additionally, the development of primary care text processing tools will reduce the need for humans to read text containing confidential information, as well as motivate the development of future research into mining information currently locked up in primary care text notes.

The Harvey Corpus, the two sets of annotation guidelines, and the work associated with creating and using these resources are another important group of contributions to the fields of NLP and corpus linguistics. Despite the corpus being available only under a special licence, the annotation and the guidelines are freely available, thus allowing reusability under certain conditions.

Finally, a more technical contribution is the concept recognition method for terse text, which was successfully applied to primary care notes, outperforming the standard method in both an idealised and an emulated realistic scenario. Even though the idea of using an entity classifier in an NER task is not new (Carreras et al., 2002; Wang and Patrick, 2009), it was previously used to correct the output of a BIO-based NER tagger. The work presented in this thesis showed that in the case of terse primary care text the first step is not necessary if all base NPs are assumed to be classification candidates.

6.4 LIMITATIONS OF THE WORK

The main limitation of this work is the difficulty in distributing its data resources. Even though models, guidelines and annotation are freely available, building on this research will be difficult unless the source data can be licensed.

The size of the annotated data is another limitation. The Harvey Corpus has more records than the average i2b2 corpus, but the mean size of GP notes is much smaller than those of other document types used in clinical corpora, e.g. discharge summaries and radiology reports. The increasing trend of the learning curve shown in Chapter 3 suggests that more annotated data should increase the performance of the machine learning models, although at a decreasing rate. Additionally, due to the limited resources available for the work described in this thesis, a compromise was made with the quality of part-of-

speech annotation. Ideally, a new specialised model should be developed for primary care text, either through training on enough in-domain annotated text, or through a domain adaptation method. In either case, improving the POS annotation of the Harvey Corpus is very likely to lead to better results in higher level tasks.

The limitations noted above have an impact on the performance of the final system, which is perhaps the only way that their influence can be objectively quantified. Thus overcoming them should also indirectly improve the system.

6.5 FUTURE WORK

There are several directions in which this research can be developed.

Given additional resources, the current annotated data could easily be expanded in different ways, including making it larger using the rest of the unannotated text notes under the same licence. The benefit of in-domain part-of-speech annotation could also be assessed through a sufficiently large annotated corpus, and extended to the whole corpus if found appropriate. It would also be interesting to investigate adapting a POS tagset to primary care data as mentioned in Chapter 4. Additionally, the approach to annotation could be further tested by comparing current inter-annotator agreement performance with linguistics expert annotators, or even teams combining linguistic and clinical expertise. The medical concept annotations could also be linked to SNOMED-CT concepts, which would allow the systems trained on the corpus to produce more generalisable results.

Domain adaptation is a direction that should certainly be explored in any further development of this work, especially if enough POS annotated data could be produced. The approach has high potential for improving POS tagging, due to the already good performance of the general domain models on primary care data. Another area that could be explored further in light of the recently published work by Levy et al. (2015) is word representation, and word embeddings in particular. Levy et al. presented an in-depth analysis of the algorithms behind word2vec and GLoVe, identifying their preprocessing and parameters to show that embeddings are produced in a similar way to positive pointwise mutual information (PPMI), which is an established method in earlier distributional semantics work (Baroni and Lenci, 2010; Turney and Pantel, 2010). The authors also demonstrated that embeddings have been thus far somewhat unfairly evaluated, as the new software

packages have been used as black boxes without accounting for some hidden processes, which could also be applied to PPMI. The word representation experiments reported in this thesis are not exhaustive as they do not optimise the embeddings creation process itself, but considering these findings, it is now evident that achieving better results using word embeddings requires more attention and thorough experimentation than previously thought.

A robust basic natural language processing framework allows the development of solutions to a variety of more complex problems, such as relation recognition. Longitudinal relations, for example, are particularly important for clinical data as they provide an overview of the data that can reveal specific slowly developing patterns (Stubbs et al., 2015b).

Negation and other elements of context around concepts also play an important role in correctly analysing the clinical data. Often the mention of a concept does not automatically mean its presence in the patient at the time of the examination. It may refer to its absence, previous occurrence, a reference to family health history, or merely the possibility of presence. It is also important to determine the identity of such concepts by linking different text representations of the same entity. There are multiple approaches to the problem that could be explored in the context of primary care text in the future. A thorough analysis should include experiments with rule-based systems, supervised machine learning models, and more complex hybrid systems.

Finally, the work described in this thesis and solutions to these and other similar problems could make future health research using primary care text faster, more scalable, and more effective, while decreasing the need for researchers to access confidential data.

A

APPENDIX A: ANNOTATION GUIDELINES I

A.1 INTRODUCTION

THE PURPOSE of these guidelines is to introduce the reader to the annotation of general practitioner (GP) notes with syntactic chunks and semantic expressions. To achieve that these guidelines provide some basic grammar and linguistics knowledge, as well as some additional instructions about annotation techniques.

THE TASK itself amounts to identifying and annotating a number of different linguistic phrases and expressions in GP notes using the web-based annotation tool **Brat**.

THE MOTIVATION for this task lies in the crafting of gold standard data for training and evaluating machine learning tools that will automate the process of linguistic analysis. This automated process will ultimately serve as the basis of more complex analysis leading to the automated extraction of information about symptoms and diseases from GP notes.

THE INFORMATION in these guidelines is distributed in three sections. The **Common Grammar** section introduces the reader to basic notions of grammar and linguistics. This section may be skipped by a reader with prior linguistic experience. The **Chunks** section explains the notion of syntactic chunks and their annotation according to these guidelines. In the last section, called **Annotation**, we discuss the details of a good annotation practice, as well as some of the specific issues and tasks of the annotation of medical records.

NOTES on the use of bold and italics in the guidelines. All examples are marked with *italics*. The focus area of the example, e.g. the phrase head, is marked with ***bold italics***. Key points in the guidelines are highlighted in **bold face** only.

A.2 COMMON GRAMMAR

This section explains the basic notions of parts of speech, noun and adjective phrases, and main verbs. This information is crucial for a complete understanding of the guidelines. However, readers who are familiar with basic grammar should be able to skip it and continue reading at Section A.3.

A.2.1 *Parts of speech*

English words are traditionally classified into eight lexical categories, or parts of speech: nouns, pronouns, adjectives, verbs, adverbs, prepositions, conjunctions, and interjections.

- **Nouns** are abstract or concrete entities: a person (*policeman*, *Michael*), a place (*bank*, *Brighton*), a real object (*tie*, *radio*), an imaginary object (*unicorn*), a feeling or an idea (*joy*, democracy), or a quality (*cleverness*). Nouns could be grouped together to form compound nouns as in *bus driver*, *desk lamp*, or *party animal*.
- **Pronouns** are generally used instead of nouns and personal names in different situations with different functionality. Here are examples of all the types: *I*, *you*, *we*, etc.; *me*, *her*, *them*, etc.; *my*, *mine*, *your*, *yours*, *their*, *our*, etc.; *this*, *that*, *these*, *those*, etc.; *anyone*, *anything*, etc.; *who*, *which*, etc.; *who*, *whom*, etc.
- **Adjectives** are words used to describe qualities and attributes of nouns: *green, lazy, tall, heavy, kind*. They also include comparative and superlative forms like *better, best, worse, worst, taller, tallest, etc.*
- **Verbs** are words that indicate an action (*walk*, *write*), an occurrence (*happen*, *occur*), or a state of being (*be*).
- **Adverbs** are qualifiers of adjectives (***slightly*** *green*; ***absolutely*** *fresh*), verbs (*to work* ***efficiently***; ***suddenly*** *disappeared*), clauses, sentences, or other adverbs (*walking* ***slightly*** *impatiently*). They are usually the answer to the questions *How?*, *Where?*, or *When?*.
- **Prepositions** are words that express some sort of relation, for example a spacial relation is expressed by prepositions such as *to*, *under*, *before*, *inside*, etc.
- **Conjunctions** are words that connect other words and phrases, e.g. *and* and *or*.
- **Interjections** are words of emotional greeting (or exclamation) like *wow*, *tut-tut*, *ugh*

A.2.2 *Phrases*

In everyday speech, an arbitrary group of words may be called a **phrase**. However, in linguistics, a phrase is defined as one word or a sequence of words that function as a single unit in the syntax of a sentence.

A.2.2.1 *Noun Phrases*

A noun phrase (NP) is a unit centred around one noun or pronoun or a gerund, which is called the **head** of the noun phrase. The rest of the phrase consists of **modifiers** that give further information about the head. So if a noun denotes an entity (e.g. *dog*), the noun phrase provides us with more information about that entity. For example, in the sentence

John has a big brown dog the noun phrase *a big brown dog* gives us information about the colour and size of a dog owned by John. Other examples of the information conveyed by modifiers are attributes (*the green mile*), location (*the door in the floor*), ownership (*my girl*), quantity (*seven samurai*), and other more complex notions (*the girl who played with fire*). Usually the entity that is in the focus of the noun phrase is a noun or a pronoun, but it could also be the present participle form of a verb (as in *I love* ***reading***), which is called a **gerund** (see Section A.2.3 for more information).

A.2.2.2 *Adjective Phrases*

Adjective phrases (AP), are syntactic constructions with a head and zero or more modifiers. The head of an adjective phrase is naturally an adjective, e.g. *very* ***fast***. The number of words in APs may vary just like the one of NPs. APs could also include some modifiers of the adjective head as in *The river is* ***crystal clear***, *wound* ***severely infected***, *The wine tastes* ***very good*** and *Patient feels* ***slightly constipated***.

A.2.3 *Verbs*

For the purpose of these guidelines we discuss three types of verbs depending on their semantic and syntactic roles in a sentence or a clause. The first group is that of the **main verbs** which express the central action, occurrence, or state of being of the sentence or the clause. The second group, which are called **auxiliary verbs**, carry additional grammatical information about the main verb such as tense, (passive) voice, or modality. It is important to note that some auxiliary verbs can be also main verbs and even be used twice in the same sentence with different function or meaning as in sentence 2. below. Some main verbs, called **phrasal verbs**, on the other hand, can be comprised of a verb and a preposition or a particle as illustrated in example 6. below. The following examples show verbs in SMALL CAPS and main verbs in **bold**:

1. *The bears* ARE **eating** *the berries in the garden.*
2. *The bears* HAVE *always* **had** *berry snacks in the summer.*
3. *The bears* WILL **eat** *berries.*
4. *The bears* CAN **eat** *berries if they* ***find*** *any.*
5. *The bears* SHOULD *not* **eat** *too many berries.*
6. *The bears* COULD **run into** *berry bushes.*

The third type of verbs, which are called **raising verbs**, always appear in conjunction with the main verb as shown (in bold) in the following sentences:

7. *The bears* **need** *to* EAT *berries.*
8. *The bears* **have** *to* EAT *berries.*
9. *The bears* **appear** *to* EAT *berries.*
10. *The bears* **seem** *to* EAT *berries.*

A.2.4 *Gerunds*

Gerunds are a special case where verbs in present participle form (ending with *-ing*) act as nouns and form noun phrases. Here are some examples of base NPs with gerunds:

11. *This house needs* ***cleaning****.*
12. *The patient has* ***normal bowel emptying****.*
13. ***Apple picking*** *is fun.*

A.3 ANNOTATION TYPES

In this section we describe the different types of annotations and we provide guidance for their correct annotation. The annotations are divided into two groups: phrase chunks and expressions. The former includes syntactic chunks based on noun phrases, adjective phrases and main verbs, while the latter includes locative, temporal, quantitative and "on-examination" expressions.

A.3.1 *Base Noun Phrase Chunks*

Base NPs are a subset of NPs that **do not contain other NPs**. A useful trick for identifying them is to watch out for prepositions, which should not be present in them. For example, ***The cat*** *in* ***the mirror*** is not a single base NP chunk, but two of them (in bold). To help identify base NPs we provide a list of modifiers that can appear within them along with examples:

- determiners
  - articles: ***the*** *dog*, ***a*** *cleaning*
  - demonstrative pronouns: ***this*** *girl*, ***that*** *man*
  - possessive determiners: ***my*** *homecoming*, ***your*** *dog*, ***our*** *car*, *the* ***police officer's*** *wife*, *the* ***neighbour's*** *constant complaining*
  - quantifiers: ***some*** *people*, ***every*** *day*, ***most*** *children*, ***any*** *student*, ***all*** *birds,* ***no*** *coffee*, ***five*** *cakes*, ***10*** *miles*
- adjectives preceding the head, such as *large*, *beautiful*, *sweeter*, *excruciating*, *soothing*
- nouns immediately preceding the head, such as *college* in *a* ***college*** *student*; note that the number of preceding nouns is unconstrained, so they could be stacked as in ***bus*** *driver*, ***school bus*** *driver*, ***city school bus*** *driver*

ANNOTATION Two things need to be considered when annotating a base NP: its head and its borders. First, there should be just one head in the NP. In most cases this means just one noun, pronoun, or a gerund inside it. For example, *a good* ***dog***, *twelve angry* ***men***, ***someone***. However, one has to be careful not to mistake the nouns modifying the head as heads. Consider the base NP *the black dragon* ***tattoo***, in which the word *dragon* modifies *tattoo*, which is the head of the base NP. The same logic is applied when dealing with more than one preceding nouns, e.g. *the school bus* ***driver***. One should also bear in mind the opposite situation, where two base NPs are listed one after another without the usual punctuation, as well as the mixture of both. Consider the sample clinical text snippet *c/o fever cough back pain*, in which there are three base NPs: *fever*, *cough*, and *back pain*. Another matter to consider is the length of base NPs, which in some cases could be quite substantial, e.g. *the long winding high mountain roads*.

A.3.2 *Adjective Chunks*

We define **adjective chunks** (AP) to be adjective phrases that act as predicative expressions. In simple terms this generally means when they follow a copula verb. The copula verbs used most often in the notes are *to be* and *feel*, although there are also others: *seem*, *look*, etc. In grammatical text copula verbs should always be present in predicative constructions (*The carpet is* ***red***; *The sky looks* ***dark***), but in the notes they are often omitted and the construction is implied by the context: *blood pressure* ***normal***; *abdomen* ***tender***; *[baby] coos and* ***alert*** *and* ***happy***. Here are some examples of typical AP cases (in bold) with comments:

1. *patient is* ***anxious*** — standard predicative construction following the verb *be*
2. *chest* ***clear*** — omitted copula verb (*be*)
3. *leg feels* ***much better*** — the determiner *much* is part of the AP
4. *finger is* ***severely infected*** — the adverb *severely is part of the AP*
5. ***worried*** *wants to be admitted today* — predicative construction (*patient is worried*) was reduced to only an adjective phrase

ANNOTATION When annotating an adjective chunks the annotator should take care to include all modifiers of the adjective (usually adverbs) as in example 3. and 4. above. The annotator should always bear in mind predicative constructions and not mistake NPs with adjective modifiers for adjective phrases *The patient has* ***high fever***.

A.3.3 *Main Verb Chunks*

Main verb chunks usually contain only the main verb. The only other words that may be included are adjacent prepositions or particles in the cases of phrasal verbs, like *show up*, *take care*, *calm down*, etc. Gerunds (see end of Section A.2.3) should not be confused with main verbs, they should be annotated as part of base noun chunks (see Section A.3.2).

ANNOTATION The annotation of a main verb is more or less a straight-forward matter as its scope is usually a single word. The cases where its annotation scope spans over other words are those of phrasal verbs that include particles (*show up*, *calm down*) and/or prepositions (*power through*). However, those particles should not be annotated when they are not adjacent to the main verb as in *Please,* ***calm*** *him* ***down***. Verb negation should not be included in the main verb chunk annotation also when it is contracted (***is****n't*).

A.3.4 *Expressions*

TEMPORAL EXPRESSIONS are words, phrases or clauses that **contain information related to time**. Some refer to a specific moment or a period in time related to an event discussed in the sentence like *in seven minutes*, *an hour ago*, *yesterday*, *next year*. Others refer to the duration of an event, for example, *for three days*, *lasting two weeks*. The third type of temporal expressions describes frequency of repetition: *twice a day*, *every week*, *biannually*, etc. Temporal expressions may not always refer to time units directly, sometimes they refer to the time or duration of other events as in *last time*, *when they were young*, *while the sun was up*. There are also temporal expressions that are more vague and indefinite like *recently* and *already*.

LOCATIVE EXPRESSIONS cover two types of expressions related to location. The first type points to the locus of a medical finding (infection, bruise, pain, etc.). The second type points to real places such as hospitals and geographical entities. The expressions may be a one of the modifiers of a base NP as in ***back*** *pain*, or a whole prepositional phrase such as ***in the hospital***. There are cases in the notes where loci are expressed with omitted or ungrammatical syntactic constructions, for example ***chest*** *pain* ***lower left quadrant***.

QUANTITATIVE EXPRESSIONS represent some sort of **quantity or measurement**, like number (*five spots*), weight (*5kg*, *ten grams*), volume (*10cc*, *a pint*), length (*12cm*, *three inches*), etc. The quantity in a quantitative expression doesn't need to be explicit, it may be vaguely defined or inferred as in *several inches* and *a few kilos*. It is important to emphasise that the items that are quantified should also be part of the annotation, i.e. *a few* ***kilos***, *12****cm***. Quantities of time like *two hours* could also be regarded as quantitative expressions, but for the purposes of annotation they should **NOT** be annotated as quantitative expressions (see Section A.4.3). There are also numbers that are not quantities, but identifiers or placements in a sequence, e.g. *group 3*, *second testing*, *phone 012345678*. Such number occurrences should not be annotated as quantities.

on-examination EXPRESSIONS are different versions of the the expression *on examination* that **marks the border between the complains and the examination observations**. These expressions are constructed and/or abbreviated in different ways, e.g. *o/e* or *during examination*. The task of the annotator is to identify such expressions in the record.

A.4 ANNOTATION PROCESS

The process of annotation is the assignment of labels, called tags, to parts of the text based on the definitions and instructions in the sections above. This section discusses some technical issues and rules of conduct for this process, common problems and a few useful annotation tips.

A.4.1 *Annotation Tasks*

The first stage of the annotation process, called **prime annotation**, is the stage when two or more annotators annotate the same data independently, assigning the annotation tags listed below. After the prime annotation process is complete, the results go through a process, called **referral**, that resolves any disagreement between the prime annotations to ensure the quality and consistency of the annotation.

LIST OF ANNOTATION TAGS:

- Noun Phrase Chunk (NP)
- Adjective Chunk (AP)
- Main Verb (MV)
- Locative Expression (LE)
- Temporal Expression (TE)
- Quantitative (QE)
- On-Examination Expression (OE)

A.4.2 *Prime Annotation Tips*

This section gives directions about specific data issues and discusses some general good annotation practices. It also gives some tips to help ensure more consistent annotation results.

We recommend that annotation is done one record at a time, considering the whole record and not just parts of it. This means that the annotator should read the whole record and try to understand its meaning before starting to annotate.

It is important to remember that lists of items of the same type should be annotated separately. For example, the Christmas presents in the following sentence are three different NP chunks: *Johnny got* ***a teddy bear***, ***a remote-controlled car***, *and* ***a hokey stick*** *for Christmas*.

If an expression seems complicated and the annotator is unsure how to deal with it, he or she can start by annotating to its left and right thus closing down its word span and making the task easier.

Annotation is not an exact science and sometimes the descriptions in the guidelines won't fit perfectly. In those cases the annotator should make an approximate decision, which is acceptable as long as it doesn't stray too far from the guidelines. The annotator should also have in mind that the decisions they make will undergo a referral process that is meant to improve the quality of annotation in exactly such ambiguous or unclear situations.

Consistency is the annotator's best friend. The annotator should make sure that they handle similar situations in the same way across the whole data. Going back to fix things is inevitable, but it gets harder towards the end of the annotation, so we recommend paying extra attention to cases that seem difficult in the beginning, when going back and fixing all previous occurrences is still feasible.

In cases of uncertainty we recommend being conservative. When none of the annotations seem to fit, no annotation should be made. Also in the cases of conjunction (see Section A.4.4), chunks should be annotated separately unless it is clear they are part of entities that should be annotated together, e.g. *bits and pieces*, *this and that*, *black and white*.

Finally, we advise the annotators to not worry too much. They shouldn't "overthink" problems as this might lead to confusion. If they encounter more difficult cases, they should look for them or a version of them here in the guidelines or just make a note of them and carry on.

A.4.3 *Priority and Embedding of Annotation*

The phrase chunks defined in these guidelines are very restricted. They have a very basic form with just a few words. We call chunks and/or expressions embedded if one of them contains the other. They coincide if they include the same words. Embedding and coinciding of annotations is **allowed when a phrase chunk or a main verb is contained in an expression** or the other way around (see 1. and 2. in Example A.1). However, while embedding phrase chunks into expressions is allowed, their **partial overlapping is not** (see 3. in Example A.1). Therefore, in order to avoid errors in identifying the span of the expressions, we recommend that the annotation of phrases and main verbs precede the annotation of expressions. The rules for embedding annotation are incorporated in **Brat** (see Section A.4.8), which warns the annotators when they make overlapping annotations by **highlighting it in red**.

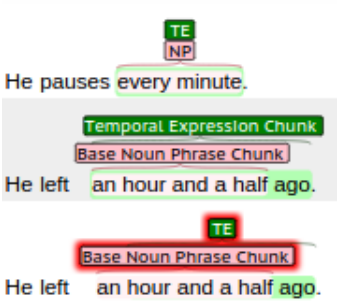


Figure A.1: Examples of annotation overlapping. NP chunks are denoted with bold face and temporal expressions are underlined

A.4.4 *Including Conjunctions*

Conjunctions are words such as *and* and *or*, which are used to join words, phrases, clauses, and sentences together. In most cases they are not included in the annotation of phrase chunks (NPs, APs, MVs), except when they connect modifiers (*the tall and handsome man*; *a green or blue jacket*; *slow but steady pace*; *scary yet exciting adventures*). Normally when conjunctions connect two phrases as in the sentence *We have* ***a cat*** *and* ***a dog***, the phrases are annotated separately. Only when they connect phrases that usually go together and have become fixed expressions like *black and white* or *bits and pieces*, they are annotated as one phrase.

However, if in doubt about whether a conjunction should be included in a phrase, it is recommended that the more conservative decision is taken, namely separate annotation.

The guidance above is not relevant to conjunctions within expressions (see Section A.3.4), which are not defined so strictly.

A.4.5 *Redacted Text*

The clinical records data contained sensitive information that was redacted and replaced with the tilde symbol (~). In most cases the context gives enough information to make a good guess what sort of words are missing – usually either names or places. Such redacted words should be annotated as if they were apparent. The annotator should also make a note of their guess about such words. For example, in *as per dr.* ~~~~~~ *advise* the doctor's name was removed, but it should still be annotated as an NP chunk. The annotators should use upper case letters with surrounding angle brackets to denote abstract entities in their guesses. For example, *<NAME>'s* will be a good guess for the redacted text from the example above.

A.4.6 *Abbreviations and Acronyms*

Abbreviations and acronyms need to be considered and annotated as their full forms. For example, the phrase *poss ovarian* should be annotated as an adjective phrase, and *FBC* (*full blood count*) should be annotated as a noun phrase. However, acronyms of phrases and sentences that could not be annotated with one tag like *spt* (*seen patient today*) should not be annotated. As a rule of thumb the annotators should think about acronyms as their full forms and annotate them accordingly **only** if the whole acronym can be annotated with one annotation.

A.4.7 *Punctuation and Special Symbols*

The nature of the clinical records data uses punctuation and special symbols in two different ways: 1. in their classical context and usage, and 2. as an abbreviation or a substitute for words and/or expressions.

Square bracket prefixes should be excluded from the annotations. For example, *[D]**Difficulty*** should be annotated as a NP-chunk starting after the closing bracket.

NORMAL PUNCTUATION

As a rule of thumb punctuation that is inside the span of an annotation should be left there (e.g. hyphens, (*on-line*), apostrophes (*Jimmy's*), commas, etc.) and punctuation that is on the annotation fringes should be excluded (e.g. quotation marks, braces, etc.). An exception to the last rule is the apostrophe sign in sentences like *The dog is **my neighbours'***.

SPECIAL SYMBOLS AND PUNCTUATION

As mentioned before, often punctuation and other symbols like a plus, a minus, slash, etc. are used for some peculiar unorthodox purposes. Annotators should try to use their best judgement in identifying the purpose of the symbols in these cases. Symbols that are used to convey the meaning of words should be treated as such. For example, question marks could replace the word *possible*, and a series of plus signs could indicate an increase in some value. We encourage the annotators to use their judgement in the annotation of such symbols, but we emphasise the need for consistency in their decisions.

A.4.8 *Brat Annotation Tool*

The annotations will be recorded using the **Brat** annotation tool, which is a web-based tool that allows annotators to access data from and input annotation to a remote server. No local installation is required, only **JavaScript** and **cookies** need to be enabled for the successful loading of the tool in the browser . The full functionality and performance of the tool is only guaranteed when using the latest version of one of the two supported browsers: **Google Chrome** or **Safari**. Even if the tool seems to run using other browsers, such as **Internet Explorer** or **Firefox**, they should not be used, because its performance there is not predictable and it may end up damaging the input or even the existing data.

PRIME ANNOTATION

The prime annotators will be assigned a personal folder with documents each containing a small batch of clinical records. The folder should be loaded using the **Collection** button in the left upper corner of the interface. Annotations are created by selecting the portion of a text using the mouse and choosing the appropriate annotation type from the pop-up menu. Annotations can be edited or deleted by double-clicking on them at any time. We recommend reading through the tutorial available at http://weaver.nlplab.org/~brat/demo/latest/#/ for a better and more practical understanding of the annotation process with Brat.

ANNOTATION REFERRAL

The annotation referral process aims at selecting the best version of an annotation out of all prime versions. The referee should edit merged versions of all annotations, exposing all conflicting annotations. They should resolve the conflicts by **deleting** the less appropriate annotations.

A.4.9 *Annotation Referee*

This section gives instructions for the **annotation referee** and should not concern prime annotators.

The most important notion that we want to emphasise here is that the annotation referee must NOT add any information, but only choose between already existing annotations without changing them. In the cases of only one existing prime annotation for a certain bit of text, the annotation should remain unchanged. If all prime annotations seem wrong, the one that seems nearest to a correct annotation among them should be chosen.

When comparing annotations of roughly the same chunks or expressions, the annotation label is more important than the annotated word sequence. For instance consider the sentence in Example A.2. If one annotation identifies the word sequence *big black bear* as an NP chunk and the other identifies the word sequence *the big black bear* as an AP chunk, the NP chunk annotation is considered better, because even though it should include one more word, it is labelled correctly as a NP chunk.

The word sequence of an annotation on the other hand is important to the referee in cases of annotations with the same label. Then the annotation that includes the word sequence closest to the correct one is considered better. For example, consider the NP chunk *the big black bear* in Example A.2, which is identified in two different ways (1. and 2.). The annotation in 2. is considered the correct (or the better) choice, because it includes all the words of the NP chunk.

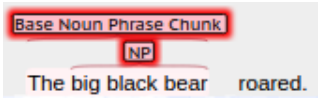


Figure A.2: Word span of annotation example

Naturally, in cases of uncertainty with regard to the word span, the annotation referee is advised to be conservative and keep to smaller word spans and the label choice of the majority (if applicable).

B

APPENDIX B: ANNOTATION GUIDELINES II

B.1 SEMANTIC ENTITIES

The purpose of these guidelines is to introduce the reader to the annotation of three types of semantic entities in primary care free text notes using the *brat* web annotation platform. The entities are defined as follows with their respective label abbreviations:

- symptoms, and signs of a disease or disorder, as well as syndromes (NPS)
- disease name or shorthand (NPD)
- drug names (NPDRG)

The guidelines assume that these entities can only be found as base noun phrases (NP) at least in the primary care text at hand. Therefore, the annotators should in fact annotate existing base NP annotation with the semantic entities listed above. A fourth type of annotation, a not-entity (NPN), is used to allow keeping track of already considered NPs.

B.2 ANNOTATION MECHANICS

The annotators will be using the *brat* annotation platform, which allows interactive labelling of free text. The data will be displayed ten notes at a time with base NP annotations. The label of each NP needs to be changed to one of the four annotations listed above. NPs that are not considered semantic entities should nonetheless be annotated with the *not-entity* label. Annotation should be performed alone without consultation between different annotators, although using literature and other resources is allowed.

B.3 DISAGREEMENT RESOLUTION

After both annotators complete the annotation process, the resulting annotations are merged and used in a new round of annotation resolution. During this round the two annotators consider all disagreeing annotations together, and determine the better solution through discussion and reference to medical literature. In the cases where diseases are also signs (symptoms) of other diseases, the *disease* label should be applied.

C

APPENDIX C: MISCELLANEOUS

*Name	Abbreviation	Description *
*Correct	COR	number correct *
*Partial	PAR	number partially correct (no par-* tial credit was given in MUC-7)
*Incorrect	INC	number incorrect *
*Missing	MIS	number missing *
*Spurious	SPU	number spurious *
*Non-committal	NON	number non-committal (null fills* generated by system that were null in the answer key)
*Possible	POS	number possible (COR + INC +* MIS), i.e. the number of fills in the answer key plus any optional fills allowed by the key and generated by the system.
*Actual	ACT	number actual (COR + INC +* SPU), i.e. the number of fills generated by the system under evaluation
*Precision	PRE	Precision = COR/POS *
*Recall	REC	Recall = COR/ACT *
*Undergeneration	UND	Undergeneration = MIS/POS *
*Overgeneration	OVG	Overgeneration = SPU/ACT *
*Substitution	SUB	Substitution = INC/(COR + INC)*
*Error per response fill	ERR	Error per response fill* = (INC + SPU + MIS)/(COR+INC+SPU+MIS)

Table C.1: MUC-7 scoring table.

```
def f1_eval(gold, inf):
    # symptoms, diseases, drug names
    tagset = {'NPS', 'NPD', 'NPDRG'}
    tp, tn, fp, fn = 0, 0, 0, 0
    for g, i in zip(gold, inf):
        if i in tagset: # positive
            if i == g:
                tp += 1 # true positive
            else:
                fp += 1 # fasle positive
        elif i != g: # negative
            fn += 1 # false negative
    pr = 100 * float(tp) / (tp + fp) if tp + fp else 0
    re = 100 * float(tp) / (tp + fn) if tp + fp else 0
    f1 = 2.0 * pr * re / (pr + re)  if tp + fp else 0
    return f1, pr, re
```

Algorithm C.1: A Python implementation of the f-score calculation for the Divide & Conquer approach where non-entity labels are considered negative.

Feature Name	**Left Border**	**Right Border**
word	-1	1
POS tag	-1	1
POS tag bigram	-1	0
POS tag trigram	-1	0
canonicalised form	-1	1
Ney-Essen clusters	-1	0
noun suffixes	-1	0
adjective suffixes	-1	0
medical suffixes	-1	0
medical prefixes	-1	0

Table C.2: Final feature vector with context windows used for chunking models.

D

APPENDIX D: TABLES & FIGURES

	size	source	Twitter	NPS	Stanford	nopos
CW	25	RCV1	-0.971	-0.798	-0.168	0.114
CW	50	RCV1	-0.877	-0.646	-0.168	0.139
CW	100	RCV1	-0.778	-0.575	-0.059	0.059
CW	200	RCV1	-0.905	-0.686	-0.074	0.046
OSCCA	200	RCV1	-0.910	-0.386	-0.128	0.196
TSCCA	200	RCV1	-0.939	-0.841	-0.241	0.005
HLBL	50	RCV1	-0.741	-0.507	-0.074	0.139
HLBL	100	RCV1	0.386	-0.721	-0.046	0.336
word2vec	25	GPRD	-0.646	0.507	0.241	0.723
word2vec	50	GPRD	0.444	-0.798	-0.139	0.296
word2vec	100	GPRD	0.575	-0.798	-0.168	0.462
word2vec	300	Google	-0.989	-0.137	-0.081	0.139

Table D.1: Significance test results for chunking models using embeddings features. The p-values were calculated using Wilcoxon signed-rank test. Negative p-values signify performance lower than the baseline.

source	corpus	size	f-score	precision	recall
-	GPRD	512	71.58	77.85	66.40
-	GPRD	1000	71.55	77.84	66.38
Stanford	RCV1	512	70.16	78.44	63.72
-	GPRD	250	70.16	77.48	64.31
-	GPRD	10000	69.34	80.67	61.06
Baseline	-	-	68.43	83.67	58.15

Table D.2: Comparison of performance impact by different Ney-Essen cluster features in DC entity recognition with positional feature sets. Baseline model uses word, POS, and preceding context word features.

Corpus	#	Document type	Annotation type
Harvey Corpus	750	GP notes	syntactical chunks, 4 semantic annotation types
Pakhomov et al. (2004)	273	outpatient notes, discharge summaries, inpatient service notes	POS tags
Uzuner et al. (2007b)	889	discharge summaries	de-identification, smoker status (subset)
Uzuner (2009)	1237	discharge summaries	present, absent, questionable for obesity + 15 comorbidities
Uzuner et al. (2010b)	1243	discharge summaries	medications, dosages, frequencies, modes, reasons, durations, list/narrative
Uzuner et al. (2011)	871	discharge summaries, progress reports	concepts, assertions, relations
Sun et al. (2013a)	310	discharge summaries	temporal relations
Roberts et al. (2009)	565K	histopathology reports, clinical narratives, and imaging reports	entities and relations
Pakhomov et al. (2004)	271	clinical notes	POS
Ogren et al. (2008)	160	outpatient notes	concepts from a subset of SNOMED-CT
Voorhees and Hersh (2012)	~17K	patient visits consisting of history and physical reports, surgical pathology reports, radiology reports	topics
Pestian et al. (2007)	1954	radiology reports	ICD-9-CM codes
Wang and Patrick (2009)	311	admission summaries	entities based on SNOMED-CT
Fan et al. (2011)	50	progress reports	POS
Fan et al. (2013)	25	progress reports	syntactic trees of ill-formed sentences

Table D.3: A non-exhaustive list of notable clinical corpora. Note that the size of GP notes is around 30 tokens, while the length of other documents varies, but is generally greater.

	size	data	Twitter	NPS	Stanford	nopos
Brown	50	Twitter	0.515	-0.575	0.508	0.005
Brown	100	PubMed	0.284	0.799	0.575	0.005
Brown	150	PubMed	0.508	-0.959	0.799	0.005
Brown	320	PubMed	0.093	0.878	0.214	0.005
Brown	500	PubMed	0.059	-0.721	0.415	0.005
Brown	1000	PubMed	0.093	0.508	0.241	0.005
Brown	100	RCV1	0.333	-0.959	0.203	0.005
Brown	320	RCV3	0.285	0.721	-0.859	0.005
Brown	1000	RCV2	0.508	-0.241	0.445	0.005
Brown	3200	RCV4	0.575	-0.444	-0.445	0.005
Brown	100	GPRD	0.541	0.333	0.445	0.005
Brown	250	GPRD	0.047	0.575	0.333	0.005
Brown	500	GPRD	0.066	0.386	0.093	0.005
Ney-Essen	512	GPRD	-0.571	0.481	0.239	0.005
Ney-Essen	512	RCV1	-0.694	0.052	0.029	0.005

Table D.4: Significance test results for chunking models using word representation cluster features. The p-values were calculated using Wilcoxon signed-rank test. Negative p-values signify performance lower than the baseline.

preceding	following	f-score	precision	recall
1	0	68.42	83.67	58.15
1	1	68.42	83.67	58.15
1	2	67.81	81.68	58.27
2	0	66.93	80.59	57.47
2	1	66.93	80.59	57.47
0	1	66.91	85.09	55.38
0	2	66.73	83.64	55.77
2	2	66.15	78.78	57.19
0	*0*	*66.91*	*85.09*	*55.38*

Table D.5: Comparing different scopes of context features for positional DC document classification of entity recognition. Baseline without any additional context is in italics.

source	corpus	size	method	f-score	precision	recall
Dhillon	RCV1	200	oscca	69.78	79.70	62.37
Dhillon	RCV1	200	tscca	69.78	79.70	62.37
Turian	RCV1	100	HLBL, scaled	68.91	79.66	60.95
Turian	RCV1	50	HLBL, scaled	68.91	79.66	60.95
Turian	RCV1	100	HLBL	68.91	79.66	60.95
Turian	RCV1	50	HLBL	68.91	79.66	60.95
Turian	RCV1	100	C&W	68.80	79.42	60.93
Turian	RCV1	200	C&W	68.80	79.42	60.93
Turian	RCV1	25	C&W	68.80	79.42	60.93
Turian	RCV1	50	C&W	68.80	79.42	60.93
Turian	RCV1	100	C&W, scaled	68.80	79.42	60.93
Turian	RCV1	200	C&W, scaled	68.80	79.42	60.93
Turian	RCV1	25	C&W, scaled	68.80	79.42	60.93
Turian	RCV1	50	C&W, scaled	68.80	79.42	60.93
-	GPRD	100	word2vec	68.56	80.80	59.83
-	GPRD	25	word2vec	68.56	80.80	59.83
-	GPRD	50	word2vec	68.56	80.80	59.83
Baseline	-	-	-	68.43	83.67	58.15

Table D.6: Comparison of performance impact by different word embeddings features in DC entity recognition with positional feature sets. Baseline model uses word, POS, and preceding context word features.

source	corpus	size	f-score	precision	recall
-	GPRD	512	72.67	79.39	67.00
Ananiadou	PubMed	1000	72.52	79.69	66.53
Ananiadou	PubMed	320	71.52	78.53	66.01
Ananiadou	PubMed	500	71.10	78.66	65.21
-	GPRD	250	71.10	77.36	66.09
-	GPRD	100	70.50	78.99	64.03
Ananiadou	PubMed	150	70.26	78.23	64.07
ARK	Tweets	1000	69.60	77.88	63.05
Turian	RCV1	3200	69.36	78.29	62.45
Turian	RCV1	1000	69.20	77.67	62.62
Turian	RCV1	320	68.97	77.25	62.45
Turian	RCV1	100	68.93	78.78	61.50
Ananiadou	PubMed	100	68.59	79.12	60.77
Baseline	-	-	68.43	83.67	58.15

Table D.7: Comparison of performance impact by different Brown cluster features in DC entity recognition with positional feature sets. Baseline model uses word, POS, and preceding context word features.

sizes	**type**	**F1**	**precision**	**recall**
(2,)	suffix	78.30	82.64	74.62
(2,3)	suffix	77.91	83.12	73.57
(3,)	suffix	77.91	83.12	73.57
(2,3,4)	suffix	77.00	82.84	72.19
(2,4)	suffix	77.00	82.84	72.19
(3,4)	suffix	77.00	82.84	72.19
(4,)	suffix	77.00	82.84	72.19
(3,)	prefix	78.26	82.12	75.06
(2,3)	prefix	78.26	82.12	75.06
(2,)	prefix	78.09	81.82	74.95
(2,3,4)	prefix	78.07	82.34	74.47
(2,4)	prefix	78.07	82.34	74.47
(3,4)	prefix	78.07	82.34	74.47
(4,)	prefix	78.07	82.34	74.47
Baseline	-	75.83	81.62	70.83

Table D.8: All experiment results with affix features using linear kernel SVM with positional features. The baseline uses words, POS, preceding context words, word bigrams, and all word representation features.

stat	fts	classifier	10%	20%	50%	90%	crafted
χ^2	pos	kNN	34.03	25.66	13.20	30.31	51.02
F-test	pos	kNN	38.15	31.94	29.10	48.42	51.02
χ^2	pos	Linear SVM	79.94	78.04	76.56	79.45	80.00
F-test	pos	Linear SVM	80.02	78.06	76.56	79.45	80.00
χ^2	pos	Decision Tree	68.52	68.79	68.86	68.98	70.54
F-test	pos	Decision Tree	68.81	68.90	69.04	69.17	70.54
χ^2	P	Naïve Bayes	78.25	79.47	80.10	71.62	67.95
F-test	P	Naïve Bayes	78.22	79.47	80.10	71.62	67.95
χ^2	pos	Polynomial SVM	77.69	75.30	72.14	76.41	75.78
F-test	pos	Polynomial SVM	77.80	75.26	72.14	76.41	75.78
χ^2	pos	Sigmoid SVM	49.91	49.82	48.67	49.41	62.07
F-test	pos	Sigmoid SVM	50.37	49.14	48.67	49.41	62.07
χ^2	P	SVM_{RBF}	79.03	80.17	62.92	42.45	45.32
F-test	P	SVM_{RBF}	78.89	80.05	62.92	42.45	45.32
χ^2	bow	Polynomial SVM	69.96	69.54	68.89	69.15	73.91
F-test	bow	Polynomial SVM	69.24	68.78	66.09	66.24	73.91
χ^2	bow	kNN	29.89	30.01	30.07	30.15	32.50
F-test	bow	kNN	29.84	30.12	30.23	30.48	32.50
χ^2	bow	Linear SVM	68.01	68.15	68.47	68.61	72.70
F-test	bow	Linear SVM	67.63	67.39	67.56	67.58	72.70
χ^2	bow	Decision Tree	59.21	59.46	59.62	59.94	60.80
F-test	bow	Decision Tree	59.12	59.34	59.67	60.01	60.80
χ^2	bow	Naïve Bayes	64.74	66.46	58.06	52.16	73.24
F-test	bow	Naïve Bayes	63.94	66.21	58.12	52.18	73.24
χ^2	bow	Sigmoid SVM	52.33	51.98	51.78	51.57	65.55
F-test	bow	Sigmoid SVM	53.01	52.74	52.12	51.79	65.55
χ^2	bow	RBF SVM	60.62	60.94	59.37	58.99	63.69
F-test	bow	RBF SVM	60.87	60.13	59.39	58.99	63.69

Table D.9: Comparison between DC models using different classifiers with automatically selected positional and BoW features (10%, 20%, 50%, and 90%), and the crafted feature set.

BIBLIOGRAPHY

A. M. Rassinoux and R. H. Baud and J. R. Scherrer (1994). A Multilingual Analyser of Medical Texts. In W. M. Tepfenhart and J. P. Dick and J. F. Sowa, editor, *Conceptual Structures: Current Practices*, pages 84–96. Springer, Berlin, Heidelberg.

A. Roberts and R. Gaizauskas and M. Hepple (2008). Extracting Clinical Relationships from Patient Narratives. In *Proceedings of the ACL Workshop on Current Trends in Biomedical Natural Language Processing*, pages 10–18, Columbus, Ohio. Association for Computational Linguistics.

A. Roberts and R. Gaizauskas and M. Hepple and Y. Guo (2008). Combining terminology resources and statistical methods for entity recognition: an evaluation. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, Marrakech.

Abney, S. (1991). Parsing by Chunks. In Robert C. Berwick, Steven P. Abney, and Carol Tenny, editors, *Principle-Based Parsing: Computation and Psycholinguistics*, pages 257–278. Kluwer, Dordrecht.

Abney, S. (1995). Chunks and Dependencies: Bringing Processing Evidence to Bear on Syntax. In *Computational Linguistics and the Foundations of Linguistic Theory*, pages 145–164. CSLI.

Agirre, E. and Martínez, D. (2004). Unsupervised WSD based on Automatically Retrieved Examples: The Importance of Bias. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 25–32. Association for Computational Linguistics.

Aizerman, M. A., Braverman, E. A., and Rozonoer, L. (1964). Theoretical foundations of the potential function method in pattern recognition learning. In *Automation and Remote Control,*, number 25 in Automation and Remote Control, pages 821–837.

Albright, D., Lanfranchi, A., Fredriksen, A., Styler, W. F., Warner, C., Hwang, J. D., Choi, J. D., Dligach, D., Nielsen, R. D., Martin, J., et al. (2013). Towards comprehensive syntactic and semantic annotations of the clinical narrative. *Journal of the American Medical Informatics Association*, 20(5):922–930.

Alex, B., Haddow, B., and Grover, C. (2007). Recognising Nested Named Entities in Biomedical Text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 65–72. Association for Computational Linguistics.

Alhelbawy, A. and Gaizauskas, R. J. (2014). Graph Ranking for Collective Named Entity Disambiguation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 75–80, Baltimore, MD, USA.

Alnazzawi, N., Thompson, P., and Ananiadou, S. (2014). Building a semantically annotated corpus for congestive heart and renal failure from clinical records and the literature. In *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis*, pages 69–74. Association for Computational Linguistics.

Ávarez, L. G., Aylin, P., Tian, J., King, C., Catchpole, M., Hassall, S., Whittaker-Axon, K., and Holmes, A. (2011). Data linkage between existing healthcare databases to support hospital epidemiology. *Journal of Hospital Infection*, 79(3):231–235.

Andersen, P. M., Hayes, P. J., Huettner, A. K., Schmandt, L. M., Nirenburg, I. B., and Weinstein, S. P. (1992). Automatic Extraction of Facts from Press Releases to Generate News Stories. In *Proceedings of the Third Conference on Applied Natural Language Processing*, ANLC 1992, pages 170–177. Association for Computational Linguistics.

Aramaki, E., Imai, T., Miyo, K., and Ohe, K. (2006). Patient Status Classification by Using Rule based Sentence Extraction and BM25 kNN-based Classifier. In *i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data*.

Arlot, S. and Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79.

Artstein, R. and Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Aw, A., Zhang, M., Xiao, J., and Su, J. (2006). A phrase-based statistical model for SMS text normalization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 33–40. Association for Computational Linguistics.

Azzalini, A. and Scarpa, B. (2012). *Data Analysis and Data Mining: An Introduction*. Oxford University Press, Inc., New York, USA.

Bada, M., Eckert, M., Evans, D., Garcia, K., Shipley, K., Sitnikov, D., Jr., W. A. B., Cohen, K. B., Verspoor, K., Blake, J. A., and Hunter, L. E. (2012). Concept annotation in the CRAFT corpus. *BMC Bioinformatics*, 13:161.

Balasuriya, D., Ringland, N., Nothman, J., Murphy, T., and Curran, J. R. (2009). Named Entity Recognition in Wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 10–18. Association for Computational Linguistics.

Baroni, M. and Lenci, A. (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

Baud, R. H., Lovid, C., and Rossinoux, A.-M. (1998). Morpho-semantic parsing of medical expressions. In *Annual Symposium of AMIA*.

Bejček, E., Hajičová, E., Hajič, J., Jínová, P., Kettnerová, V., Kolářová, V., Mikulová, M., Mírovský, J., Nedoluzhko, A., Panevová, J., Poláková, L., Ševčíková, M., Štěpánek, J., and Zikánová, Š. (2013). Prague Dependency Treebank 3.0.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. (2010). A Theory of Learning from Different Domains. *Machine Learning*, 79(1-2):151–175.

Bennett, E. M., Alpert, A., and Goldsein, A. C. (1954). Communications Through Limited Questioning. *Public Opinion Quarterly*, 18(3):303–308.

Bentley, T., Price, C., and Brown, P. (1996). Structural and lexical features of successive versions of the Read Codes. In *Proceedings of the Annual Conference of The Primary Health Care Specialist Group of the British Computer Society*, page 91–103.

Bharati, A., Sangal, R., Sharma, D. M., and Bai, L. (2006). ANnCorra: Annotating corpora guidelines for POS and chunk annotation for Indian languages. Technical Report TR-LTRC-31, LTRC, IIIT-Hyderabad.

Bies, A., Ferguson, M., Katz, K., MacIntyre, R., Tredinnick, V., Kim, G., Marcinkiewicz, M. A., and Schasberger, B. (1995). Bracketing Guidelines for Treebank II Style Penn Treebank Project. Technical report, University of Pennsylvania.

Bikel, D. M. (2002). Design of a Multi-lingual, Parallel-processing Statistical Parsing Engine. In *Proceedings of the Second International Conference on Human Language Technology Research*, pages 178–182, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Black, E., Jelinek, F., Lafferty, J., Magerman, D. M., Mercer, R., and Roukos, S. (1992). Towards History-based Grammars: Using Richer Models for Probabilistic Parsing. In *Proceedings of the Workshop on Speech and Natural Language*, pages 134–139. Association for Computational Linguistics.

Bohnet, B. (2010). Very High Accuracy and Fast Dependency Parsing is Not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97. Association for Computational Linguistics.

Bohnet, B. and Nivre, J. (2012). A Transition-based System for Joint Part-of-speech Tagging and Labeled Non-projective Dependency Parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1455–1465. Association for Computational Linguistics.

Boisen, S., Crystal, M., Schwartz, R. M., Stone, R., and Weischedel, R. M. (2000). Annotating Resources for Information Extraction. In *Second International Conference on Language Resources and Evaluation*. European Language Resources Association.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press.

Bottou, L. (2001). *Une Approche théorique de l'Apprentissage Connexioniste; Applications á la reconnaissance de la Parole.* PhD thesis, Université Paris XI.

Bresnan, J. (2001). *Lexical-Functional Syntax*. Blackwell Publishers, Oxford, UK.

Brill, E. and Moore, R. (2000). An Improved Error Model for Noisy Channel Spelling Correction. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong.

Brin, S. (1999). Extracting Patterns and Relations from the World Wide Web. In *Selected Papers from the International Workshop on The World Wide Web and Databases*, WebDB 1998, pages 172–183, London, UK, UK. Springer-Verlag.

Briscoe, T., Carroll, J., and Watson, R. (2006). The second release of the RASP system. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 77–80. Association for Computational Linguistics.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based N-gram Models of Natural Language. *Computational Linguistics*, 18(4):467–479.

Buscaldi, D., Rosso, P., Pla, F., Segarra, E., and Arnal, E. S. (2006). Verb Sense Disambiguation Using Support Vector Machines: Impact of Wordnet-extracted Features. In *Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 192–195, Berlin, Heidelberg. Springer-Verlag.

Buzhou, T., Xuan, W., and Xiaolong, W. (2008). Chunking with Max-Margin Markov Networks*. In *22nd Pacific Asia Conference on Language, Information and Computation*, pages 474–480.

Carletta, J. (1996). Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, 22:249–254.

Carreras, X., Màrquez, L., and Padró, L. (2002). Named Entity Extraction Using AdaBoost. In *Proceedings of the 6th Conference on Natural Language Learning*, volume 20 of *COLING 2002*, pages 1–4, Stroudsburg, PA, USA. Association for Computational Linguistics.

Carrero, F. M., Gomez Hidalgo, J. M., Puertas, E., M., M., and Mata, J. (2006). Quick prototyping of high performance text classifiers. In *i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data*.

Carroll, J., Koeling, R., and Puri, S. (2012). Lexical acquisition for clinical text mining using distributional similarity. In *Proceedings of the 13th international conference on Computational Linguistics and Intelligent Text Processing*, pages 232–246, Berlin, Heidelberg. Springer-Verlag.

Caruana, R. and Freitag, D. (1994). Greedy Attribute Selection. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 28–36. Morgan Kaufmann.

Chapman, W. W., Bridewell, W., Hanbury, P., Cooper, G. F., and Buchanan, B. G. (2001). A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries. *Journal of Biomedical Informatics*, 2001:34–301.

Charniak, E. (2000). A Maximum-entropy-inspired Parser. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, pages 132–139. Association for Computational Linguistics.

Charniak, E. and Johnson, M. (2005). Coarse-to-fine N-best Parsing and MaxEnt Discriminative Reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 173–180. Association for Computational Linguistics.

Chen, D. and Manning, C. (2014). A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Chen, Q., Li, H., Tang, B., Liu, X., Liu, Z., Liu, S., and Wang, W. (2014). Identifying risk factors for heart disease over time – HITSZ's system for track 2 of the 2014 i2b2 NLP challenge. In *Seventh i2b2 Shared Task and Workshop: Challenges in Natural Language Processing for Clinical Data*, Washington, DC.

Chinchor, N. (1998). MUC-7 Test Scores Introduction. In *Proceedings of the Seventh Message Understanding Conference*.

Choi, J. D. and Palmer, M. (2010). Robust constituent-to-dependency conversion for English. In *9th International Workshop on Treebanks and Linguistic Theories*, pages 55–66. Association for Computational Linguistics.

Chomsky, N. (1957). *Syntactic Structures*. Mouton, The Hague.

Choudhury, M., Saraf, R., Jain, V., Sarkar, S., and Basu, A. (2007). Investigation and modeling of the structure of texting language. In *In Proceedings of the IJCAI-Workshop on Analytics for Noisy Unstructured Text Data*, pages 63–70.

Church, K. and Gale, W. (1991). Probability Scoring for Spelling Correction. *Statistics and Computing*, 1(2): 93–103.

Church, K. W. (1988). A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, ANLC '88, pages 136–143. Association for Computational Linguistics.

Clark, A. (2003). Combining distributional and morphological information for part of speech induction. *Proceedings of 10th Meeting of the European Chapter of the Association for Computational Linguistics*, pages 59–66.

Clark, A. (2013). Learning Trees from Strings: A Strong Learning Algorithm for Some Context-free Grammars. *The Journal of Machine Learning Research*, 14(1):3537–3559.

Clear, J. H. (1993). The British National Corpus. In Landow, G. P. and Delany, P., editors, *The Digital Word*, pages 163–187. MIT Press, Cambridge, MA, USA.

Coden, A., Pakhomov, S., Ando, R., Duffy, P., and Chute, C. (2005). Domain-specific language models and lexicons for tagging. *Journal of Biomedical Informatics*, 38:422–430.

Cohen, A. M. (2008). Case report: Five-way smoking status classification using text hot-spot identification and error-correcting output codes. *JAMIA*, 15(1):32–35.

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46.

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220.

Cohen, K. B., Lanfranchi, A., Corvey, W., Baumgartner Jr., W. A., Roeder, C., Ogren, P. V., Palmer, M., Hunter, and Lawrence (2010). Annotation of all coreference in biomedical text: Guideline selection and adaptation. In *BioTxtM 2010: 2nd Workshop on Building and Evaluating Resources for Biomedical Text Mining*, pages 37–41.

Cohen, P. R. (1995). *Empirical Methods for Artificial Intelligence*. MIT Press, Cambridge, MA, USA.

Collins, M. (2003). Head-Driven Statistical Models for Natural Language Parsing. *Computational Linguistics*, 29(4):589–637.

Collobert, R. and Weston, J. (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167, New York, NY, USA. ACM.

Cook, P. and Stevenson, S. (2009). An unsupervised model for text message normalization. In *CALC '09: Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pages 71–78. Association for Computational Linguistics.

Corbett, P. T. and Copestake, A. A. (2008). Cascaded classifiers for confidence-based chemical named entity recognition. *BMC Bioinformatics*, 9(S-11).

Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3):273–297.

Covington, M. A. (2001). A fundamental algorithm for dependency parsing. In *Proceedings of the 39th Annual ACM Southeast Conference*, pages 95–102.

Crammer, K. and Singer, Y. (2002). On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. *Journal of Machine Learning Research*, 2:265–292.

Culler, J. D. (1976). *Saussure*. Harvester Press.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Dang, H. T. and Palmer, M. (2002). Combining Contextual Features for Word Sense Disambiguation. In *Proceedings of the ACL '02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, pages 88–94. Association for Computational Linguistics.

Daumé, III, H., Kumar, A., and Saha, A. (2010). Frustratingly Easy Semi-supervised Domain Adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 53–59. Association for Computational Linguistics.

Daumé III, H. (2007). Frustratingly Easy Domain Adaptation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic.

Davies, M. and Fleiss, J. L. (1982). Measuring agreement for multinomial data. *Biometrics*, 38(4):1047–1051.

de Cruys, T. V. (2010). A non-negative tensor factorization model for selectional preference induction. *Natural Language Engineering*, 16(4):417–437.

De Marneffe, M.-C. and Manning, C. D. (2008). The Stanford Typed Dependencies Representation. In *COLING 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, CrossParser '08, pages 1–8. Association for Computational Linguistics.

Derczynski, L., Ritter, A., Clark, S., and Bontcheva, K. (2013). Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data. In Angelova, Galia and Bontcheva, Kalina and Mitkov, Ruslan, editor, *RANLP*, pages 198–206. RANLP.

Dhillon, P., Foster, D., and Unger, L. (2015). Eigenwords: Spectral Word Embeddings. *Journal of Machine Learning Research*, 16. (To Appear).

Diab, M. and Resnik, P. (2002). An Unsupervised Method for Word Sense Tagging Using Parallel Corpora. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 255–262. Association for Computational Linguistics.

Dorr, D. A., Phillips, W. F., Phansalkar, S., Sims, S. A., and Hurdle, J. F. (2006). Assessing the difficulty and time cost of de-identification in clinical narratives. *Methods of information in medicine*, 45(3):246–252.

Dridan, R. and Oepen, S. (2012). Tokenization: Returning to a Long Solved Problem a Survey, Contrastive Experiment, Recommendations, and Toolkit. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers*, pages 378–382. Association for Computational Linguistics.

Driver, H. and Kroeber, A. (1932). *Quantitative Expression of Cultural Relationship*, volume Quantitative Expression of Cultural Relationships of *Publications in American Archaeology and Ethnology*. Berkeley. University of California.

Efron, B. (1979). Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, 7:1–26.

Efron, B. (1983). Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. *Journal of the American Statistical Association*, 78(382):316–331.

Efron, B. and Tibshirani, R. (1997). Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560.

Engel, J. (1988). Polytomous logistic regression. *Statistica Neerlandica*, 42(4):233–252.

Erk, K. (2012). Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 10(6):635–653.

Estivill-Castro, V. (2002). Why So Many Clustering Algorithms: A Position Paper. *SIGKDD Explorations Newsletter*, 4(1):65–75.

Evert, S. (2005). *The Statistics of Word Co-Occurrences: Word Pairs and Collocations*. PhD thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, Stuttgart, Germany.

Fan, J.-W., Prasad, R., Yabut, R. M., Loomis, R. M., Zisook, D. S., Mattison, J. E., and Huang, Y. (2011). Part-of-speech Tagging for Clinical Text: Wall Or Bridge Between Institutions? In *American Medical Informatics Association Annual Symposium*, 1, pages 382–391. American Medical Informatics Association.

Fan, J.-W., Yang, E., Jiang, M., Prasad, R., Loomis, R., Zisook, D., Denny, J., Xu, H., and Huang, Y. (2013). Research and applications: Syntactic parsing of clinical text: guideline and corpus development with handling ill-formed sentences. *JAMIA*, 20(6):1168–1177.

Fellbaum, C. (2005). WordNet and wordnets. In Brown, Keith, editor, *Encyclopedia of Language and Linguistics*, pages 665–670, Oxford. Elsevier.

Feller, W. (1968). *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, 3rd edition.

Ferraro, J. P., III, H. D., DuVall, S. L., Chapman, W. W., Harkema, H., and Haug, P. J. (2013). Improving performance of natural language processing part-of-speech tagging on clinical narratives through domain adaptation. *JAMIA*, 20(5):931–939.

Fielstein, E. M., Brown, S. H., and Speroff, T. (2004). Algorithmic De-identification of VA Medical Exam Text for HIPAA Privacy Compliance: Preliminary Findings. *Medinfo*.

Finkel, J. R., Kleeman, A., and Manning, C. D. (2008). Efficient, Feature-based, Conditional Random Field Parsing. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 959–967, Columbus, Ohio. Association for Computational Linguistics.

Finkel, J. R. and Manning, C. D. (2009). Hierarchical Bayesian Domain Adaptation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 602–610. Association for Computational Linguistics.

Firth, J. (1957). *Papers in linguistics, 1934-1951*. Oxford University Press.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Ford, E., Carroll, J., Smith, H., Davies, K., Koeling, R., Petersen, I., Rait, G., and Cassell, J. (under revision 2015). Rheumatoid arthritis in UK general practice – what can free text tell us about diagnostic recording? *BMJ Open*.

Ford, E., Nicholson, A., Koeling, R., Tate, A., Carroll, J., Axelrod, L., Smith, H. E., Rait, G., Davies, K., Petersen, I., Williams, T., and Cassell, J. (2013). Optimising the use of electronic health records to estimate the incidence of rheumatoid arthritis in primary care: What information is hidden in free text? *BMC Medical Research Methodoclogy*, 13(105):1–12.

Forsythand, E. N. and Martell, C. H. (2007). Lexical and Discourse Analysis of Online Chat Dialog. In *Proceedings of the International Conference on Semantic Computing*, ICSC '07, pages 19–26, Washington, DC, USA. IEEE Computer Society.

Foster, J. (2007). Treebanks gone bad: Parser evaluation and retraining using a treebank of ungrammatical sentences. *International Journal on Document Analysis and Recognition*, 10(3):129–145.

Frederick, P. D. (2003). Linkage of Patient Registries and Clinical Data Sets without Patient Identifiers. In *SUGI 28 Proceedings*.

Gale, W., Church, K., and Yarowsky, D. (1993). A method for disambiguating word senses in a large corpus . Technical report, AT&T Bell Laboratories.

Galley, M. and McKeown, K. (2003). Improving Word Sense Disambiguation in Lexical Chaining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 1486–1488, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Giménez, J. and Màrquez, L. (2004). SVMTool: A general POS tagger generator based on support vector machines. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, Lisbon, Portugal.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 42–47. Association for Computational Linguistics.

Grefenstette, G. (1994). *Explorations in automatic thesaurus discovery*. Springer.

Grishman, R. and Sundheim, B. (1996). Message Understanding Conference-6: A Brief History. In *Proceedings of the 16th Conference on Computational Linguistics*, pages 466–471. Association for Computational Linguistics.

Grover, C. and Tobin, R. (2006). Rule-based Chunking and Reusability. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*.

Guillen, R. (2006). Automated De-identification and Categorization of Medical Records. In *i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data*.

Guyon, I. and Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3:1157–1182.

Haenel, V., Gouillart, E., and Varoquaux, G. (2013). SciPy Lecture Notes. http://scipy-lectures.github.io/_images/svm_margin.png.

Halgrim, S., Xia, F., Solti, I., Cadag, E., and Uzuner, O. (2010). Extracting Medication Information from Discharge Summaries. In *Proceedings of the HLT/NAACL 2010 Second Louhi Workshop on Text and Data Mining of Health Documents*, pages 61–67. Association for Computational Linguistics.

Han, B., Cook, P., and Baldwin, T. (2012). Automatically Constructing a Normalisation Dictionary for Microblogs. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 421–432, Jeju Island, Korea.

Hatori, J., Miyao, Y., and Tsujii, J. (2008). Word sense disambiguation for all words using tree-structured conditional random fields. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 43–46.

Heinze, D. T., Morsch, M. L., Potter, B. C., and Sheffer, R. E. (2008). Medical i2b2 NLP Smoking Challenge: The A-Life System Architecture and Methodology. *Journal of the American Medical Informatics Association : JAMIA*, 15(1):40–43.

Hirschman, L., Colosimo, M., Morgan, A., and Yeh, A. (2005). Overview of BioCreAtIvE task 1B: normalized gene lists. *BMC Bioinformatics*, 6 Suppl 1.

Hirschman, L., Grishman, R., and Sager, N. (1976). From Text to Structured Information: Automatic Processing of Medical Reports. In *Proceedings of the June 7-10, 1976, National Computer Conference and Exposition*, AFIPS '76, pages 267–275, New York, NY, USA. ACM.

Hobbs, J. R. (1993). The Generic Information Extraction System. In *Proceedings of the 5th Conference on Message Understanding*, MUC5 '93, pages 87–91. Association for Computational Linguistics.

Hobbs, J. R. (2002). Information Extraction from Biomedical Text. *J. of Biomedical Informatics*, 35(4):260–264.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). OntoNotes: The 90% Solution. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 57–60. Association for Computational Linguistics.

Hripcsak, G. and Rothschild, A. S. (2005). Technical Brief: Agreement, the F-Measure, and Reliability in Information Retrieval. *JAMIA*, 12(3):296–298.

Hsu, L. M. and Field, R. (2003). Inter-rater agreement measures: Comments on Kappa[n], Cohen's Kappa, Scott's Pi, and Aickin's Alpha. *Understanding Statistics*, 2:205–219.

Huang, F. and Yates, A. (2009). Distributional Representations for Handling Sparsity in Supervised Sequence-labeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 495–503. Association for Computational Linguistics.

ISO (2008). ISO DIS 24617-1: 2008 Language Resource Management - Semantic Annotation Framework - Part 1: Time and Events. Technical report, International Standards Organisation.

Jason Eisner, editor (2007). *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, Prague, Czech Republic.

Jiang, M., Huang, Y., Fan, J.-w., Tang, B., Denny, J. C., and Xu, H. (2014). Parsing Clinical Text: How Good Are the State-of-the-Art Parsers? In *Proceedings of the ACM 8th International Workshop on Data and Text Mining in Bioinformatics*, DTMBIO '14, pages 1–1, New York, NY, USA. ACM.

Jindal, P. and Roth, D. (2013). End-to-End Coreference Resolution for Clinical Narratives. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*.

Jindal, P., Roth, D., and Gunter, C. A. (2014). Joint Inference for End-to-end Coreference Resolution for Clinical Notes. In *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 192–201, New York, NY, USA. ACM.

Joshi, A. K. and Schabes, Y. (1997). Tree-adjoining Grammars. In Rozenberg, Grzegorz and Salomaa, Arto, editor, *Handbook of Formal Languages, Vol. 3*, pages 69–123. Springer-Verlag New York, Inc., New York, NY, USA.

Joshi, M., Pakhomov, S. V. S., Pedersen, T., and Chute, C. G. (2006a). A comparative study of supervised learning as applied to acronym expansion in clinical reports. In *AMIA 2006, American Medical Informatics Association Annual Symposium, Washington, DC, USA, November 11-15, 2006*.

Joshi, M., Pedersen, T., Maclin, R., and Pakhomov, S. V. S. (2006b). Kernel Methods for Word Sense Disambiguation and Acronym Expansion. In *AAAI*, pages 1879–1880. AAAI Press.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Kilgarriff, A. (2001). Comparing Corpora. *International Journal of Corpus Linguistics*, 6(1): 1–37.

Kilgarriff, A. and Yallop, C. (2000). What's in a Thesaurus? In *Proceedings of the Second Conference on Language Resources and Evaluation*, pages 1371–1379.

King, L. (1982). *Medical thinking: a historical preface*. Princeton University Press, Princeton, N.J.

Klein, D. and Manning, C. D. (2003). Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430. Association for Computational Linguistics.

Kobus, C., Yvon, F., and Damnati, G. (2008). Normalizing SMS: are Two Metaphors Better than One? In *Proceedings of the Conference on Computational Linguistics*, pages 441–448.

Koeling, R. (2000). Chunking with Maximum Entropy Models. In *Proceedings of the 2Nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning*, ConLL '00, pages 139–141. Association for Computational Linguistics.

Koeling, R., Carroll, J., Tate, R., and Nicholson, A. (2011a). Annotating a corpus of clinical text records for learning to recognize symptoms automatically. In Nytrø, Ø., Slaughter, L., and Moen, H., editors, *LOUHI 2011 Third International Workshop on Health Document Text Mining and Information Analysis. CEUR Workshop Proceedings*, pages 43–50, Trondheim, Norway.

Koeling, R., Tate, A. R., and Carroll, J. A. (2011b). Automatically estimating the incidence of symptoms recorded in GP free text notes. In *Proceedings of the first international workshop on Managing interoperability and complexity in health systems*, pages 43–50, New York, NY, USA. ACM.

Kohavi, R. (1995). A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1137–1143, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Krallinger, M., Leitner, F., Rodriguez-Penagos, C., and Valencia, A. (2008). Overview of the protein-protein interaction annotation extraction task of BioCreative II. *Genome biology*, 9 Suppl 2(Suppl 2):S4+.

Krippendorff, K. (2004). *Content Analysis: An introduction to its methodology*. Sage, Thousand Oaks, CA, 2nd edition.

Krippendorff, K. (2011). Computing Krippendorff's Alpha-Reliability. http://repository.upenn.edu/asc_papers/43.

Krippendorff, K. H. (1980). *Content Analysis: An Introduction to Its Methodology*. Sage Publications, Inc, Thousand Oaks, CA, 1st edition.

Kudo, T. and Matsumoto, Y. (2001). Chunking with support vector machines. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.

Kudo, T. and Matsumoto, Y. (2003). Fast methods for kernel-based text analysis. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 24–31. Association for Computational Linguistics.

Kuhns, R. J. (1988). A News Analysis System. In *Proceedings of the 12th Conference on Computational Linguistics*, pages 351–355. Association for Computational Linguistics.

Kukich, K. (1992). Techniques for Automatically Correcting Words in Text. *ACM Comput. Surv.*, 24(4):377–439.

Kulick, S., Bies, A., Liberman, M., Mandel, M., McDonald, R., Palmer, M., Schein, A., and Ungar, L. (2004a). Integrated Annotation for Biomedical Information Extraction. In *Proceedings of HLT/NAACL 2004 Workshop: Biolink*, pages 61–68.

Kulick, S., Bies, A., Liberman, M., Mark, M., Winters, S., and White, P. (2004b). Integrated Annotation for Biomedical Information Extraction. In *Proceedings of the workshop on text mining, ontologies and natural language processing in biomedicine*.

Kullback, S. (1935). On the Bernoulli distribution. *Bulletin of the American Mathematical Society*, 41(12):857–864.

Kučera, H. and Francis, W. N. (1967). *Computational Analysis of Present-Day American English*. Brown University Press, Providence, RI.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lavergne, T., Cappé, O., and Yvon, F. (2010). Practical very large scale CRFs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 504–513. Association for Computational Linguistics.

Lazaridou, A., Bruni, E., and Baroni, M. (2014). Is this a wampimuk? cross-modal mapping between distributional semantics and the visual world. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1403–1414, Baltimore, Maryland. Association for Computational Linguistics.

Leacock, C., Miller, G. A., and Chodorow, M. (1998). Using Corpus Statistics and WordNet Relations for Sense Identification. *Computational Linguistics*, 24(1):147–165.

Lebret, R., Legrand, J., and Collobert, R. (2013). Is Deep Learning Really Necessary for Word Embeddings? Technical report, Idiap. Accepted to NIPS Deep Learning Workshop.

Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., and Jurafsky, D. (2011). Stanford's Multi-pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34. Association for Computational Linguistics.

Leitner, F. and Krallinger, M. (2010). The FEBS Letters SDA corpus: A collection of protein interaction articles with high quality annotations for the BioCreative II.5 online challenge and the text mining community. *FEBS Letters*, 584(19):4129–4130.

Lesk, M. (1986). Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pages 24–26, New York, NY, USA. ACM.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225.

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5:361–397.

Li, C., Ji, L., and Yan, J. (2015). Acronym Disambiguation Using Word Embedding. In *AAAI Conference on Artificial Intelligence*.

Liang, P. (2005). Semi-supervised learning for natural language. Master's thesis, Department of Electrical Engineering and Computer Science, MIT.

Lita, L., Ittycheriah, A., Roukos, S., and Kambhatla, N. (2003). tRuEcasIng. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Japan.

Malouf, R. (2002). A Comparison of Algorithms for Maximum Entropy Parameter Estimation. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING 2002, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Manning, C. (2006). Doing Named Entity Recognition? Don't optimise for F_1. http://nlpers.blogspot.co.uk/2006/08/doing-named-entity-recognition-dont.html.

Manning, C. D. (2011). Part-of-speech Tagging from 97% to 100%: Is It Time for Some Linguistics? In *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 171–189, Berlin, Heidelberg. Springer-Verlag.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Markowski, C. A. and Markowski, E. P. (1990). Conditions for the Effectiveness of a Preliminary Test of Variance. *The American Statistician*, 44(4):322+.

Martin, L., Battistelli, D., and Charnois, T. (2014). Symptom extraction issue. In *Proceedings of BioNLP 2014*, pages 107–111, Baltimore, Maryland. Association for Computational Linguistics.

Martschat, S. (2013). Multigraph Clustering for Unsupervised Coreference Resolution. In *ACL (Student Research Workshop)*, pages 81–88. The Association for Computer Linguistics.

Mays, E., Damerau, F. J., and Mercer, R. L. (1991). Context based spelling correction. *Information Processing & Management*, 27(5):517–522.

Mccarthy, D., Koeling, R., Weeds, J., and Carroll, J. (2004). Using automatically acquired predominant senses for word sense disambiguation. In *Proceedings of the ACL SENSEVAL-3 workshop*, pages 151–154.

McDonald, R. and Pereira, F. (2006). Online learning of approximate dependency parsing algorithms. In *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics*, volume 6, pages 81–88.

McDonald, R., Pereira, F., Ribarov, K., and Hajič, J. (2005). Non-projective Dependency Parsing Using Spanning Tree Algorithms. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530. Association for Computational Linguistics.

McEnery, T. and Hardie, A. (2012). *Corpus Linguistics*. Cambridge University Press.

Meystre, S. M., Óscar Ferrández, Friedlin, F. J., South, B. R., Shen, S., and Samore, M. H. (2014). Text de-identification for privacy protection: A study of its impact on clinical text information content. *Journal of Biomedical Informatics*, 50:142–150. Special Issue on Informatics Methods in Medical Privacy.

Meystre, S. M., Savova, G. K., Kipper-Schuler, K. C., and Hurdle, J. F. (2008). Extracting information from textual documents in the electronic health record: a review of recent research. *Yearbook of Medical Informatics*, pages 128–44.

Mihalcea, R. (2007). Using Wikipedia for Automatic Word Sense Disambiguation. In Sidner, Candace L. and Schultz, Tanja and Stone, Matthew and Zhai, ChengXiang, editor, *Proceedings of Human Language Technologies: The 2007 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 196–203. The Association for Computational Linguistics.

Mihalcea, R. and Moldovan, D. I. (1999). An Automatic Method for Generating Sense Tagged Corpora. In Hendler, Jim and Subramanian, Devika, editor, *AAAI/IAAI*, pages 461–466. AAAI Press / The MIT Press.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

Miller, Tristan and Biemann, Chris and Zesch, Torsten and Gurevych, Iryna (2012). Using Distributional Similarity for Lexical Expansion in Knowledge-based Word Sense Disambiguation. In *Proceedings of the Conference on Computational Linguistics*, pages 1781–1796.

Mnih, A. and Hinton, G. E. (2008). A Scalable Hierarchical Distributed Language Model. In *Advances in Neural Information Processing Systems*, pages 1081–1088.

Mockus, J. (1977). On Bayesian Methods for Seeking the Extremum and their Application. In *IFIP Congress*, pages 195–200.

Montague, R. (1974). Universal Grammar. In Thomason, Richmond H., editor, *Formal Philosophy: Selected Papers of Richard Montague*, pages 222–247. Yale University Press, New Haven, London.

Moon, S., McInnes, B., and Melton, G. B. (2015). Challenges and Practical Approaches with Word Sense Disambiguation of Acronyms and Abbreviations in the Clinical Domain. *Healthcare Informatics Research*, 21(1):35–42.

Morgan, A., Lu, Z., Wang, X., Cohen, A., Fluck, J., Ruch, P., Divoli, A., Fundel, K., Leaman, R., Hakenberg, J., Sun, C., Liu, H. H., Torres, R., Krauthammer, M., Lau, W., Liu, H., Hsu, C. N., Schuemie, M., Cohen, K. B., and Hirschman, L. (2008). Overview of BioCreative II gene normalization. *Genome Biology*, 9(Suppl 2):S3+.

Napoles, C., Gormley, M., and van Durme, B. (2012). Annotated English Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction & Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100, Montreal, Canada.

Navigli, R. (2009). Word sense disambiguation: a survey. *ACM Computing Surveys*, 41(2):1–69.

Navigli, R. and Velardi, P. (2005). Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 27(7):1075–1086.

Ney, H., Essen, U., and Kneser, R. (1994). On Structuring Probabilistic Dependencies in Stochastic Language Modelling. *Computer Speech and Language*, 8:1–38.

Ng, A. Y. and Jordan, M. I. (2002). On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In T.G. Dietterich and S. Becker and Z. Ghahramani, editor, *Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press.

Ng, H. T., Wang, B., and Chan, Y. S. (2003). Exploiting Parallel Texts for Word Sense Disambiguation: An Empirical Study. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 455–462. Association for Computational Linguistics.

Nicholson, A., Ford, E., Davies, K., Smith, H. E., Rait, G., Tate, A. R., Petersen, I., and Cassell, J. (2013). Optimising use of electronic health records to describe the presentation of rheumatoid arthritis in primary care: a strategy for developing code lists. *PLoS One*, 2(8).

Nivre, J. (2003). An Efficient Algorithm for Projective Dependency Parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT*, pages 149–160.

Nivre, J., Hall, J., and Nilsson, J. (2006). Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*.

Noreen, E. W. (1989). *Computer-Intensive Methods for Testing Hypotheses : An Introduction*. Wiley-Interscience.

O'Dell, M. and Russell, R. (1922). U.S. Patent 1,435,663. U.S. Patent Office, Washington, D.C.

Ogren, P. V., Savova, G. K., and Chute, C. G. (2008). Constructing Evaluation Corpora for Automated Clinical Named Entity Recognition. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*. European Language Resources Association.

Ohta, T., Tateisi, Y., and Kim, J.-D. (2002). The GENIA corpus: an annotated research abstract corpus in molecular biology domain. In *Proceedings of the second international conference on Human Language Technology Research*, pages 82–86, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Okazaki, N. (2007). CRFsuite: a fast implementation of Conditional Random Fields (CRFs). http://www.chokkan.org/software/crfsuite/.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of HLT/NAACL*, pages 380–390.

Pakhomov, S. (2002). Semi-supervised Maximum Entropy Based Approach to Acronym and Abbreviation Normalization in Medical Texts. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 160–167. Association for Computational Linguistics.

Pakhomov, S., Coden, A., and Chute, C. (2004). Creating a Test Corpus of Clinical Notes Manually Tagged for Part-of-speech Information. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*, JNLPBA'04, pages 62–65. Association for Computational Linguistics.

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

Pantel, P. and Lin, D. (2002). Discovering Word Senses from Text. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 613–619, New York, NY, USA. ACM.

Patrick, J. and Li, M. (2009). A cascade approach to extracting medication events. In *In: Proceedings of Australasian Language Technology Workshop*, pages 10–1136.

Pedersen, T. (2001). A Decision Tree of Bigrams is an Accurate Predictor of Word Sense. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, pages 1–8. Association for Computational Linguistics.

Pedersen, T. (2002). Evaluating the Effectiveness of Ensembles of Decision Trees. In *Proceedings of the ACL '02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, pages 81–87. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Pestian, J. P., Brew, C., Matykiewicz, P., Hovermale, D. J., Johnson, N., Cohen, K. B., and Duch, W. (2007). A Shared Task Involving Multi-label Classification of Clinical Free Text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 97–104. Association for Computational Linguistics.

Petrov, S., Das, D., and McDonald, R. (2012). A Universal Part-of-Speech Tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation.*

Poesio, M., Ponzetto, S., and Versley, Y. (2011). Computational models of anaphora resolution: A survey. *Linguistic Issues in Language Technology.*

Pollard, C. and Sag, I. (1994). *Head-Driven Phrase Structure Grammar.* Chicago University Press, Chicago, Illinois.

Pollock, J. J. and Zamora, A. (1984). Automatic spelling correction in scientific and scholarly text. *Commununications of the ACM*, 27(4):358–368.

Proctor, Paul, editor (1978). *Longman Dictionary of Contemporary English.* Longman.

Pyysalo, S., Ginter, F., Moen, H., Salakoski, T., and Ananiadou, S. (2013). Distributional Semantics Resources for Biomendical Text Processing. In *Proceedings of the 5th International Symposium on Languages in Biology and Medicine*, pages 39–43, Tokyo, Japan.

R. Gaizauskas and H. Harkema and M. Hepple and A. Setzer (2006). Task-Oriented Extraction of Temporal Information: The Case of Clinical Narratives. In *Proceedings of the 13th International Symposium on Temporal Representation and Reasoning (TIME2006)*, pages 188–195.

Ramshaw, L. and Marcus, M. (1995). Text Chunking using Transformation-Based Learning. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94.

Ratnaparkhi, A. (1997). A Linear Observed Time Statistical Parser Based on Maximum Entropy Models. In *Second Conference on Empirical Methods in Natural Language Processing*, pages 1–10.

Rish, I. (2001). An Empirical Study of the Naive Bayes Classifier. Technical report, IMB.

Ritter, A., Cherry, C., and Dolan, B. (2010). Unsupervised Modeling of Twitter Conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180. Association for Computational Linguistics.

Ritter, A., Clark, S., Mausam, and Etzioni, O. (2011). Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics.

Roberts, A., Gaizauskas, R., Hepple, M., Demetriou, G., Guo, Y., and Setzer, A. (2008). Semantic Annotation of Clinical Text: The CLEF Corpus. In *Proceedings of LREC 2008 Workshop on Building and Evaluating Resources for Biomedical Text Mining*, pages 19–26, Marrakech.

Roberts, A., Gaizauskas, R. J., Hepple, M., Demetriou, G., Guo, Y., Roberts, I., and Setzer, A. (2009). Building a semantically annotated corpus of clinical texts. *Journal of Biomedical Informatics*, 42(5):950–966.

Roberts, K., Shooshan, S. E., Rodriguez, L., Abhyankar, S., Kilicoglu, H., and Demner-Fushman, D. (in press 2015). The role of fine-grained annotations in supervised recognition of risk factors for heart disease from EHRs. *Journal of Biomedical Informatics*.

Roger Garside and Geoffrey Leech and Anthony McEnery (1997). *Corpus Annotation*. Longman, London.

Rogot, E. and Goldberg, I. D. (1966). A proposed index for measuring agreement in test-retest studies. *Journal of Chronic Diseases*, 19(9):991–1006.

Ruch, P., Baud, R., and Geissbühler, A. (2003). Using Lexical Disambiguation and Named-entity Recognition to Improve Spelling Correction in the Electronic Patient Record. *Artificial Intelligence in Medicine*, 29(1-2):169–184.

Ruch, P., Baud, R. H., Rassinoux, A.-M., Bouillon, P., and Robert, G. (2000). Medical document anonymization with a semantic lexicon. In *Proceedings of the AMIA Annual Symposium*, pages 729–733.

Sager, N., Friedman, C., and Lyman, M. S. (1987). *Medical Language Processing: Computer Management of Narrative Data*. Addison-Wesley, Reading, Mass.

Santorini, B. (1990). Part-Of-Speech Tagging Guidelines for the Penn Treebank Project (3rd revision, 2nd printing). Technical report, Department of Linguistics, University of, Philadelphia, PA, USA.

Sarkar, A. (2011). Syntax: a survey of syntactic parsing. In *Multilingual Natural Language Processing Applications: From Theory to Practice*. Addison-Wesley.

Savkov, A., Carroll, J., and Cassell, J. (2014). Chunking Clinical Text Containing Non-canonical Language. In *Proceedings of the 13th BioNLP Workshop*, Baltimore, USA.

Savkov, A., Carroll, J., Koelling, R., and Cassell, J. (2016). Annotating Patient Clinical Records with Syntactic Chunks and Named Entities. *Language Reources and Evaluation*.

Savova, G., Clark, C., Zheng, J., Cohen, K. B., Murphy, S., Wellner, B., Harris, D., Lazo, M., Aberdeen, J., Hu, Q., Chute, C., and Hirschman, L. (2008a). The Mayo/MITRE system for discovery of obesity and its comorbidities. In *i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data*.

Savova, G., Masanz, J., Ogren, P., Zheng, J., Sohn, S., Kipper-Schuler, K., and Chute, C. (2010). Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.

Savova, G. K., Coden, A. R., Sominsky, I. L., Johnson, R., Ogren, P. V., Groen, P. C. d., and Chute, C. G. (2008b). Word Sense Disambiguation Across Two Domains: Biomedical Literature and Clinical Notes. *J. of Biomedical Informatics*, 41(6):1088–1100.

Schnabel, T. and Schütze, H. (2014). FLORS: fast and simple domain adaptation for part-of-speech tagging. *TACL*, 2:15–26.

Schütze, H. (1998). Automatic Word Sense Discrimination. *Computational Linguistics*, 24(1):97–123.

Scott, W. A. (1955). Reliability of content analysis: The case of nominal scale coding. *Public Opinion Quarterly*, 19(3):321–325.

Sha, F. and Pereira, F. (2003). Shallow Parsing with Conditional Random Fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 134–141. Association for Computational Linguistics.

Shah, A., Martinez, C., and Hemingway, H. (2012). The freetext matching algorithm: a computer program to extract diagnoses and causes of death from unstructured text in electronic health records. *BMC Medical Informatics & Decision Making*, 12:88.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27(3):379–423.

Shinyama, Y. and Sekine, S. (2004). Named Entity Discovery Using Comparable News Articles. In *Proceedings of the 20th International Conference on Computational Linguistics*. Association for Computational Linguistics.

Sibanda, T. and Uzuner, Ö. (2006). Role of Local Context in Automatic Deidentification of Ungrammatical, Fragmented Text. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 65–73. Association for Computational Linguistics.

Siegel, S. and Castellan, N. J. J. (1988). *Nonparametric Statistics for the Behavioral Sciences*, chapter 9.8. McGraw-Hill, New York, 2nd edition.

Simov, K., Osenova, P., and Slavcheva, M. (2004). BTB-TR03: BulTreeBank morphosyntactic tagset. Technical report, Institute for Information Technology and Communication, Bulgarian Academy of Sciences.

Singh, S., Siddiqui, T. J., and Sharma, S. K. (2014). Naïve Bayes Classifier for Hindi Word Sense Disambiguation. In *Proceedings of the 7th ACM India Computing Conference*, COMPUTE '14, pages 1:1–1:8, New York, NY, USA. ACM.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. In *Proceedings of Annual Conference on Neural Information Processing Systems*, pages 2960–2968.

Soanes, Catherine and Stevenson, Angus, editor (2003). *Oxford Dictionary of English*. Oxford University Press.

Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, 27(4):521–544.

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Sproat, R., Black, A. W., Chen, S. F., Kumar, S., Ostendorf, M., and Richards, C. (2001). Normalization of non-standard words. *Computer Speech & Language*, 15(3):287–333.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012a). brat: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics.

Stenetorp, P., Soyer, H., Pyysalo, S., Ananiadou, S., and Chikayama, T. (2012b). Size (and domain) matters: Evaluating semantic word space representations for biomedical text. In *Proceedings of the 5th International Symposium on Semantic Mining in Biomedicine*, Zürich, Switzerland.

Stevenson, M., Guo, Y., Al Amri, A., and Gaizauskas, R. (2009). Disambiguation of Biomedical Abbreviations. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 71–79. Association for Computational Linguistics.

Stigler, S. M. (1983). Who Discovered Bayes's Theorem? *The American Statistician*, 37(4a):290–296.

Stubbs, A., Kotfila, C., and Uzuner, Ö. (in press 2015a). Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/uthealth shared task track 1. *Journal of Biomedical Informatics*.

Stubbs, A., Kotfila, C., Xu, H., and Uzuner, Ö. (in press 2015b). Identifying risk factors for heart disease over time: Overview of 2014 i2b2/uthealth shared task track 2. *Journal of Biomedical Informatics*.

Stubbs, A. and Uzuner, Ö. (in press 2015a). Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/uthealth corpus. *Journal of Biomedical Informatics*.

Stubbs, A. and Uzuner, Ö. (in press 2015b). Annotating risk factors for heart disease in clinical narratives for diabetic patients. *Journal of Biomedical Informatics*.

Suárez, A. and Palomar, M. (2002). A Maximum Entropy-based Word Sense Disambiguation System. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1–7. Association for Computational Linguistics.

Sun, W., Rumshisky, A., and Uzuner, Ö. (2013a). Annotating temporal information in clinical narratives. *Journal of Biomedical Informatics*, 46:5–12.

Sun, W., Rumshisky, A., and Uzuner, O. (2013b). Evaluating temporal relations in clinical text: 2012 i2b2 Challenge. *Journal of the American Medical Informatics Association*, 20(5):806–813.

Sun, W., Rumshisky, A., and Uzuner, O. (2013c). Temporal reasoning over clinical text: the state of the art. *Journal of the American Medical Informatics Association*, 20(5):814–819.

Sweeney, L. (1996). Replacing personally-identifying information in medical records, the scrub system. *Journal of the American Medical Informatics Association*.

Taira, R. K., Bui, A. A., and Kangarloo, H. (2002). Identification of patient name references within medical documents using semantic selectional restrictions. In *Proceedings of AMIA Symposium 2002*, pages 757–761.

Tanabe, L., Xie, N., Thom, L., Matten, W., and Wilbur, W. J. (2005). GENETAG: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics*, 6(S-1).

Tanabe, L. K. and Wilbur, W. J. (2002). Tagging gene and protein names in biomedical text. *Bioinformatics*, 18(8):1124–1132.

Tate, A. R., Martin, A. G. R., Murray-Thomas, T., Anderson, S. R., and Cassell, J. (2009). Determining the date of diagnosis – is it a simple matter? The impact of different approaches to dating diagnosis on estimates of delayed care for ovarian cancer in UK primary care. *BMC Medical Research Methodology*, 9.

Taylor, A., Marcus, M., and Santorini, B. (2003). The Penn Treebank: An Overview.

Temnikova, I., Jr., W. A. B., Hailu, N. D., Nikolova, I., Mcenery, T., Kilgarriff, A., Angelova, G., and Cohen, K. B. (2014). Sublanguage Corpus Analysis Toolkit: a Tool for Assessing the Representativeness and Sublanguage Characteristics of Corpora. In Nicoletta Calzolari (Conference Chair) and Khalid Choukri and Thierry Declerck and Hrafn Loftsson and Bente Maegaard and Joseph Mariani and Asuncion Moreno and Jan Odijk and Stelios Piperidis, editor, *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Tjong Kim Sang, E. F. and Buchholz, S. (2000). Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning*, pages 127–132. Association for Computational Linguistics.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT/NAACL 2003*, pages 142–147. Association for Computational Linguistics.

Tolentino, H. D., Matters, M. D., Walop, W., Law, B., Tong, W., Liu, F., Fontelo, P. A., Kohl, K., and Payne, D. C. (2007). A umls-based spell checker for natural language processing in vaccine safety. *BMC Medical Informatics & Decision Making*, 7:3.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich Part-of-speech Tagging with a Cyclic Dependency Network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics*

on Human Language Technology, pages 173–180. Association for Computational Linguistics.

Toutanova, K. and Manning, C. D. (2000). Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-speech Tagger. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics*, pages 63–70. Association for Computational Linguistics.

Toutanova, K. and Moore, R. (2002). Pronunciation Modeling for Improved Spelling Correction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Trust, W. (2015). Enabling Data Linkage to Maximise the Value of Public Health Research Data: full report. Technical report, Public Health Research Data Forum. http://www.wellcome.ac.uk/stellent/groups/corporatesite/@policy_communications/documents/web_document/wtp059017.pdf.

Tsuruoka, Y., Tateishi, Y., Kim, J.-D., Ohta, T., McNaught, J., Ananiadou, S., and Tsujii, J. (2005). Developing a Robust Part-of-speech Tagger for Biomedical Text. In *Proceedings of the 10th Panhellenic Conference on Advances in Informatics*, PCI'05, pages 382–392, Berlin, Heidelberg. Springer-Verlag.

Tsuruoka, Y. and Tsujii, J. (2005). Bidirectional Inference with the Easiest-first Strategy for Tagging Sequence Data. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 467–474. Association for Computational Linguistics.

Tsuruoka, Y., Tsujii, J., and Ananiadou, S. (2009). Fast Full Parsing by Linear-chain Conditional Random Fields. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 790–798. Association for Computational Linguistics.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word Representations: A Simple and General Method for Semi-supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics.

Turney, P. D. and Pantel, P. (2010). From Frequency to Meaning: Vector Space Models of Semantics. *CoRR*, abs/1003.1141.

Ushioda, A. (1996). Hierarchical Clustering of Words and Application to NLP Tasks. In *Fourth Workshop on Very Large Corpora*, pages 28–41.

Uzuner, Ö. (2009). Recognising Obesity and Comorbidities in Sparse Data. *Journal of the American Informatics Association*, 16(4):561–570.

Uzuner, Ö., Bodnari, A., Shen, S., Forbush, T., Pestian, J., and South, B. (2012). Evaluating the state of the art in coreference resolution for electronic medical records. *Journal of the American Informatics Association*, 19:786–791.

Uzuner, Ö., Goldstein, I., Luo, Y., and Kohane, I. (2007a). Patient Smoking Status from Medical Discharge Records. *Journal of the American Informatics Association*.

Uzuner, Ö., Luo, Y., and Szolovits, P. (2007b). Evaluating the state-of-the-art in automatic de-identification. *Journal of the American Medical Informatics Association : JAMIA*, 14(5):550–563.

Uzuner, Ö., Solti, I., and Cadag, E. (2010a). Extracting medication information from clinical text. *Journal of the American Informatics Association*, 17(5):514–518.

Uzuner, Ö., Solti, I., Xia, F., and Cadag, E. (2010b). Community annotation experiment for ground truth generation for the i2b2 medication challenge. *JAMIA*, 17(5):519–523.

Uzuner, Ö., South, B. R., Shen, S., and DuVall, S. L. (2011). 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Informatics Association*, 18(5):552–556.

Van Deemter, K. and Kibble, R. (2000). On Coreferring: Coreference in MUC and Related Annotation Schemes. *Computational Linguistics*, 26(4):629–637.

Vapnik, V. N. (1998). *Statistical learning theory*. Wiley, 1st edition.

Verspoor, K., Cohen, K. B., Lanfranchi, A., Warner, C., Johnson, H. L., Roeder, C., Choi, J. D., Funk, C., Malenkiy, Y., Eckert, M., Xue, N., Jr., W. A. B., Bada, M., Palmer, M., and Hunter, L. E. (2012). A corpus of full-text journal articles is a robust evaluation tool for revealing differences in performance of biomedical natural language processing tools. *BMC Bioinformatics*, 13:207.

Vilain, M. and Day, D. (2000). Phrase Parsing with Rule Sequence Processors: an Application to the Shared CoNLL Task. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 160–162.

Voorhees, E. M. and Hersh, W. (2012). Overview of the TREC 2012 Medical Records Track. In *Text REtrieval Conference 2012 Proceedings*.

Wagner, P. (2012). Machine Learning with OpenCV2. http://www.bytefish.de/pdf/machinelearning.pdf.

Walker, N. (2015). Limitations of de-identification: no reason not to share data. In *Big Data in Medicine: Exemplars and Opportunities in Data Science*.

Wang, X. and Carroll, J. (2005). Word Sense Disambiguation Using Sense Examples Automatically Acquired from a Second Language. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 547–554. Association for Computational Linguistics.

Wang, X. and Martinez, D. (2006). Word Sense Disambiguation Using Automatically Translated Sense Examples. In *Proceedings of the International Workshop on Cross-Language Knowledge Induction*, CrossLangInduction 2006, pages 45–52. Association for Computational Linguistics.

Wang, Y. and Patrick, J. (2009). Cascading Classifiers for Named Entity Recognition in Clinical Notes. In *Proceedings of the Workshop on Biomedical Information Extraction*, WBIE 2009, pages 42–49. Association for Computational Linguistics.

Warner, C., Bies, A., Brisson, C., and Mott, J. (2004). Addendum to the Penn Treebank II Style Bracketing Guidelines: Biomedical Treebank Annotation. Technical report, University of Pennsylvania.

Wei, C.-H., Peng, Y., Leaman, R., Davis, A. P., Mattingly, C. J., Li, J., Wiegers, T. C., and Lu1, Z. (2016). Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task. *The Journal of Biological Databases and Curation*.

Welch, B. L. (1947). The Generalization of 'Student's' Problem when Several Different Population Variances are Involved. *Biometrika*, 34(1/2):28–35.

Wellner, B., Huyck, M., Mardis, S., Aberdeen, J., Morgan, A., Peshkin, L., Yeh, A., Hitzeman, J., and Hirschman, L. (2007). Rapidly retargetable approaches to de-identification in medical records. *Journal of the American Medical Informatics Association : JAMIA*, 14(5):564–573.

Wikipedia (2015a). Naive Bayes Classifier. https://en.wikipedia.org/wiki/Naive_Bayes_classifier.

Wikipedia (2015b). Support Vector Machines. https://upload.wikimedia.org/wikipedia/commons/thumb/b/b5/Svm_separating_hyperplanes_%28SVG%29.svg/220px-Svm_separating_hyperplanes_%28SVG%29.svg.png.

Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83.

Wittgenstein, L. (2010). *Philosophical Investigations*. John Wiley & Sons.

Wong, W., Liu, W., and Bennamoun, M. (2006). Integrated scoring for spelling error correction, abbreviation expansion and case restoration in dirty text. In *Proceedings of the fifth Australasian conference on Data mining and analystics*, AusDM '06, pages 83–89, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Wu, Y., Xu, J., Zhang, Y., and Xu, H. (2015). Clinical Abbreviation Disambiguation Using Neural Word Embeddings. In *Proceedings of BioNLP 2015*.

Wynne, Martin, editor (2005). *Developing Linguistic Corpora: a Guide to Good Practice*. AHDS.

Xu, H., AbdelRahman, S., Jiang, M., Fan, J.-w., and Huang, Y. (2011). An Initial Study of Full Parsing of Clinical Text Using the Stanford Parser. In *Proceedings of the 2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops*, BIBMW '11, pages 607–614, Washington, DC, USA. IEEE Computer Society.

Xu, H., Stetson, P. D., and Friedman, C. (2009). Methods for building sense inventories of abbreviations in clinical notes. *Journal of the American Medical Informatics Association : JAMIA*, 16(1):103–108.

Y. Guo and R. Gaizauskas and I. Roberts and G. Demetriou and M. Hepple (2006). Identifying Personal Health Information Using Support Vector Machines. In *Proceedings of the AMIA 2006 Workshop on Challenges in Natural Language Processing for Clinical Data*, Washington.

Yamada, H. and Matsumoto, Y. (2003). Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT 03)*.

Yang, H. and Garibaldi, J. (in press 2015). Automatic detection of protected health information from clinic narratives. *Biomedical Informatics*. i2b2 NLP supplement.

Yeh, A. (2000). More Accurate Tests for the Statistical Significance of Result Differences. In *Proceedings of the 18th Conference on Computational Linguistics*, pages 947–953. Association for Computational Linguistics.

Zhang, P. (1993). Model selection via multifold cross validation. *The Annals of Statistics*, 21(1):299–313.

Zheng, J., Chapman, W. W., Crowley, R. S., and Savova, G. K. (2011). Coreference resolution: A review of general methodologies and applications in the clinical domain. *Journal of Biomedical Informatics*, 44(6):1113 – 1122.

Zwick, R. (1988). Another look at interrater agreement. *Psychological Bulletin*, 103(3):374–378.