University of Sussex

DETECTING DECEPTION USING INTERVIEW ASSISTIVE TECHNOLOGY

COLIN ASHBY

Submitted for the degree of Doctor of Philosophy
University of Sussex
November 2021

SUPERVISORS:
David Weir
Thomas Ormerod

DECLARATION

I hereby declare that this thesis has not been and will not be submitted in whole or in part to another University for the award of any other degree.

Signature:

Colin Ashby

ACKNOWLEDGEMENTS

Creating a thesis does not happen overnight, so I take this opportunity to thank all those who have supported me along the way.

First of all, Sarah, without your encouragement and support I would not have started or finished this research!

Thanks to all the TAG team, especially David Weir and Julie Weeds for sage advice and preventing me falling down too many rabbit holes. Also, Miro Batchkarov, Nestor Chavana, Justin Crow, Roland Davis, Chris Inskip, Thomas Kehrenburg, Thomas Kober, Jack Pay, Andy Robertson, David Spence, Oliver Thomas and Simon Wibberley for various assistance over the years. Also, the TAG lab purse for funding my over-zealous use of AWS during experiments.

Thanks to Team Tom, especially Tom Ormerod and Diane Sweeney for invaluable assistance during our interviewing study and also Rachael Bond for extended use of her servers. Also Justin Crow again, Miri Zilka and all other participants for taking part in our study.

Finally, this work would have no meaning without the contributions of Rutger Sound and Heinz-Harald Midgum; always remember Mothers Pride when you look out at the night sky!

ABSTRACT

This thesis presents the design, implementation and evaluation of an application designed to support interviewers in detecting deception. This application is evaluated in a job interviewing study using novice interviewers, which shows it to be a highly effective method of deception detection, correctly identifying 68.8% of deceivers overall, an increase of 107% and 97% over two baselines without application support, while reducing false positives.

We follow work that suggests effective test questioning is the key to detecting deception in interviewing. The rationale behind this approach is that a good breadth and depth of questioning increases cognitive load in deceivers, which greatly increases the chance of eliciting detectable behaviour change indicative of deception. Our application is based on Controlled Cognitive Engagement (CCE).

Our motivation for supporting interviewers is the difficulty of the interviewing task. Interviewers must simultaneously manage the interview process, observe and control the interviewee while generating probing test questions for subjects they potentially know little or nothing about.

The application developed in this thesis, called Intek, for Interview Technology, is designed to assist interviewers in generating test questions and providing checkable answers, while also providing a basis to keep track of interview progress. The information supplied by Intek aims to provide unexpected tests of expected knowledge relevant to the specific personal information provided in a CV or elicited during an interview.

Intek uses multiple information extraction pipelines, from multiple data sources, driven by state-of-the-art Natural Language Processing (NLP) techniques, such as BART for abstractive summarisation, spaCy for fast and accurate Named Entity Recognition (NER) and BERT fine-tuned on the CoNLL-2003 NER dataset for slower but best accuracy NER. These pipelines integrate into a single simple user interface which may be used by an interviewer for real-time questioning.

While most of the underlying NLP technology we used was "off the shelf", we discovered an opportunity to investigate a novel approach to web named entity recognition using HTML tags. Our Text+Tags approach resulted in F1 improvements of between 0.9% and 13.2% over a collection of five datasets and two NER models. Our approach is suitable for extracting named entities from websites containing varying amounts of HTML structure, as well as applicable to other NLP tasks.

PUBLICATION AND COLLABORATION

Most of the work detailed in Chapter 4 has been published in Ashby and Weir (2020) and was entirely my work, supervised by Professor David Weir.

Both Professor Tom Ormerod and Diane Sweeney were involved in the Intek development detailed in Chapter 3 as part of the "expert group", establishing high-level requirements and evaluating prototypes. Both were also involved in the study detailed in Chapter 5, in the design of the study itself, the delivery of condition-specific training, feedback to interviewers and the processing, checking and coherent lie-making in supplied CVs. Chapter 5 is concerned with the analysis of the Intek condition of our study; Diane Sweeney's work in the upcoming thesis Sweeney (2022) looks at the performance of the CCE condition. Apart from these collaborative contributions, all work in this thesis is my own.

CONTENTS

1 INTRODUCTION

In this chapter we introduce the problem of deception and discuss existing methods for detecting deception. We describe controlled cognitive engagement (CCE), a promising method for detecting deception, on which our application is based. We explain why detecting deception in interviewing is a cognitively demanding task. We then highlight the lack of technological support for interviewers in detecting deception and introduce our Intek application to address this. We introduce a novel method for web named entity recognition (NER) that emerged from our work on Intek. We then describe how Intek is evaluated in an empirical job interviewing study. Finally, we list the contributions to knowledge of the work in this thesis and summarise the thesis structure.

1.1 THE COST OF DECEPTION

Deception in job applicants' curriculum vitae is widespread. Levashina and Campion (2007) found 90% of job candidates "fake" information to some degree, while between 28% and 75% fake seriously in order to deceive. Faking seriously in this context involved constructing, inventing or borrowing experiences or accomplishments, or ommitting information to protect image. Henle et al. (2019) found that, from a sample of 196 undergraduate students at multiple U.S. universities, 55% of CVs contained errors, while 31% included "misrepresentations that are purposely designed to mislead recruiters". In the academic world, Phillips et al. (2019) found 44% of applications for faculty positions at a large university listed at least one unverifiable or inaccurate publication.

The events of September 11 2001 demonstrate the cost of failing to detect deception in the airline passenger security and public safety arenas. By the end of 2012 the U.S. had spent over $1 billion on behaviour detection officers (USGAO, 2011) to try and address this.

Within job interviewing, while lower-level embellishments and honest impression management are to be expected, complete fabrications can lead to potentially lethal consequences (Gibbons, 2020). The fin-

ancial cost of deception to industry is considerable. Henle et al. (2019) observe that CV deception predicts reduced job performance and increased workplace deviance. Both these factors may lead to a variety of costs, such as lower productivity both individually and as a team as a result of mis-management and lowered morale, costs associated with the performance management process and even damage to an organisation's reputation. If an employee needs to be replaced, as has happened in some high-profile cases due to CV deception (Abrams, 2014), costs of administration, covering the post and recruitment selection and training will be incurred. Costs and impact rise with salary and responsiblity. To give some financial context to these costs, in the U.S., voluntary employee turnover cost $617 billion in 2018. This figure breaks down into many factors, but just over 2% was specifically due to culture-employee misfit.

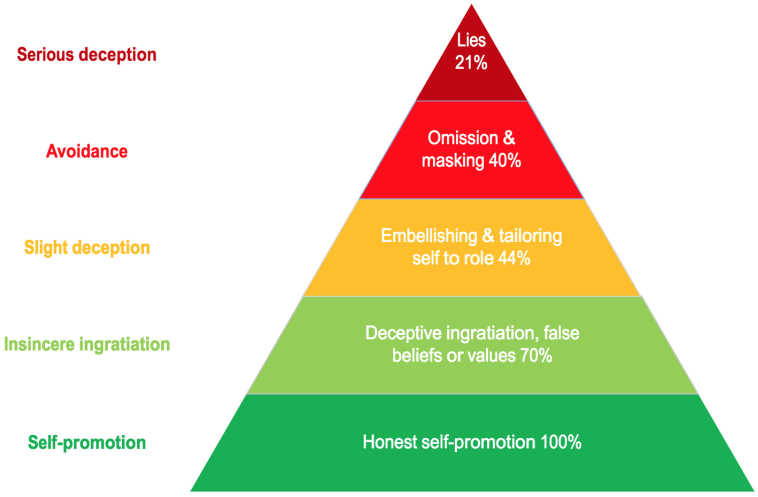

Figure 1.1: Levels of deception in real job interviews conducted in several recruitment agencies. Results combine applicant self-report with interviewer perception (Roulin et al., 2014).

1.2 STANDARD METHODS IN DECEPTION DETECTION

The primary focus of many job interviews is not usually deception detection, but rather competence in the role and organisational "fit". The majority of deception detection research is based on identifying physical or verbal behaviours that have previously been indicative of deception, such as lack of eye contact or nervousness. These approaches have a success rate of around chance (~50%). Bond Jr and DePaulo (2006) find a detection rate of 54% in a large meta-analysis, however Levine et al. (2010) suggests the performance above chance

is due to the effect of poor liars in these studies. Levine (2010) suggest the main reasons for this chance performance are poor indicators of deception, mis-placed beliefs in these indicators by interviewers, interviewer truth bias and the missing of inaccuracies in verbal accounts by interviewers. Reinhard et al. (2013) show that deception detection performance is not improved in experienced interviewers, as although they have improved beliefs about deception indicators, this improvement is counter-balanced by increased truth bias. Levine (2015) reviews approaches which show improved detection deception performance using interviewers with subject knowledge. This expertise allows improved diagnostic questioning and veracity or plausibility checking against knowledge. However, contextual expertise is not always available, especially in interviewing scenarios with heterogeneous topics, for example a general recruitment agency. These methods are discussed further in Chapter 2. The controlled cognitive engagement method was created to address these performance deficiencies without expert knowledge.

1.3 CONTROLLED COGNITIVE ENGAGEMENT METHOD

Ormerod and Dando (2015) present the controlled cognitive engagement method (CCE). CCE came about as a reaction to the poor deception detection performance of aviation security screening techniques based on predictable scripted questions and pre-defined behavioural indicators.

CCE uses evidence supplied by an interviewee to construct unexpected tests of expected knowledge that challenge an interviewee's account, using a style of questioning that controls and encourages appropriate interviewee discourse. The aim of CCE is to maximise the cognitive load for a deceiver, while appearing effortless and "normal" to a truth-teller. This cognitive load should manifest in interviewee behaviour different to that in the baseline taken at the start of the interview.

CCE has been shown to perform well in an aviation security scenario, where it detected 66% of deceptive passengers, compared with 5% for the widely-used "suspicious signs" behavioural indicator recognition method. CCE's feature of appearing "friendly and informal" to truth-tellers makes it a suitable method for integration into job interviewing.

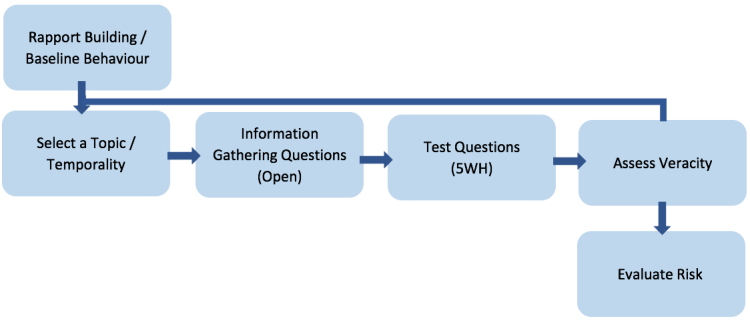

Figure 1.2: A high-level view of the iterative structure of a CCE interview.

A CCE interview follows the iterative process shown in Figure 1.2. Initially a phase of rapport building takes place, which should include at least one neutral test question. This phase includes the pleasantries of personal introduction, with the additional purpose of allowing the interviewer to build an impression of "normal" behaviour for that interviewee. The main iterative part of the interview now commences with open questioning. At some point the interviewer must choose a topic to follow (work, skills, education, interests, etc.), which may require further open information gathering questions in order to gather material to create test questions. Several test questions may then be asked, usually a maximum of five, which will ideally comprise of a variety of different temporalities (past, present, future), question styles (5WH questions: Who; What; When; Where; Why; How) and purposes. Interviewers are discouraged from directly fact checking answers through internet search, rather they look for changes in the quality and quantity of answers as well behaviour change against the initial baseline. This phase will then iterate a small number of times until the interviewer is able to make an overall evaluation of the interviewee. Job interviewing is CV-based, which allows the preparation of some test questions in advance.

This brings us onto the main rationale for the development of our interviewer support application. Interviewing is a cognitively demanding task; interviewers must simultaneously formulate questions, actively listen, respond to and veracity check answers, observe for behaviour change and keep track of the interview as a whole. It is not surprising that Levine et al. (2010) found one of the main reasons for poor deception dectection was interviewers missing inaccuracies in verbal accounts. Interviewers currently lack the ability to veracity check new information received in an interview without major disruption to the interview, while scenarios that do allow pre-

paration of test questions before the interview rely on extensive and time consuming research and collation. CCE lends us a good framework for providing technological interviewer support that can aid in the preparation, organisation and delivery of test questions in real-time, while keeping interviewers "on script" within the interview as a whole. Supporting these interviewer tasks may lower interviewer cognitive load, allowing them to focus more fully on their observation responsibilities and thereby more accurately detect deception. A wider choice of test questions of different types, delivered by a support tool, should aid the interviewer in applying the CCE principle of question variety. This variety should make deceivers' fabrication more cognitively demanding to maintain and thereby elicit behaviour change indicative of deception.

1.4 EXISTING RESEARCH IN TECHNOLOGICAL SUPPORT FOR DECEPTION DETECTION INTERVIEWING

The majority of research on technological support for detecting deception in interviewing has focussed on automatically detecting cues of deception, both linguistic (Enos, 2009; Levitan et al., 2018) and behavioural (Crockett et al., 2020; Twyman et al., 2018; O'Shea et al., 2018). With CCE we observe that behavioural cues are typically not helpful in detecting deception (Bond Jr and DePaulo, 2006; Vrij and Granhag, 2012) or not practical in a friendly job interview (DePaulo et al., 2003). CCE also theorises and observes in practice (Ormerod and Dando, 2015) that increasing the cognitive load of a deceiver through good questioning elicits behaviour change that is detectable by an interviewer over a baseline. Rather than focus on supporting deception detection per se, we instead seek to support the generation of excellent test questions, tailored to interviewee information and responses, which thereby increases the probability of eliciting behaviour change noticable by an interviewer.

The majority of existing technological support for interview questioning supplies questions that are tailored toward general knowledge of a topic or role and minimally, if at all, tailored toward the individual being interviewed, or deception detection. This clearly makes question preparation much easier, as few question variants are required. However, pre-prepared generic questions are less useful for deception detection, as they are usually expected and therefore preparable by interviewees. Also, general knowledge is less use-

ful in deception detection than questions that probe detailed episodic knowledge (Ormerod and Dando, 2015).

Technologies related to interviewing and question generation, such as computer assisted interviewing, automatic question generation and natural language generation as well as the many existing web resources for pre-defined interview questions are discussed further in the related work Chapter 2.

None of the research discussed deals with supporting interviewers to generate a good range of test questions for deception detection. Approaches that do adapt to interviewee responses, do so in a generic or naive way, not with the level of detail required to probe episodic-like knowledge.

This lack of research in real-time dynamic information extraction for interviewer support in deception detection, gives us a good opportunity. Firstly, to build and evaluate a proof of concept that does support interviewers in this way. Secondly, to gather research data on the effectiveness of our application's intervention into deception detection interviewing

1.5 OVERVIEW OF THE INTEK SYSTEM

In this section we briefly describe Intek, our application to support interviewers in deception detection interviewing.

Intek is designed to support the CCE interviewing methodology by presenting information concerning a given topic that an interviewer can use to form test questions.

Intek supports six topic types: Home; Organisation; Job; Tool; Course; Interest. These topics correspond to the main types of CV items discussed in interviews. In this thesis we always capitalise topic types to distinguish them.

The Intek search function requires a single topic type selection and a single "input snippet" of information which are both used to drive a personalised search. Table 1.1 shows example Input Snippets for each topic type.

Topic	Example Snippets
Home	RH12 1TR, Plano Texas
Organisation	Easyjet, University of Sussex, Horsham Cycling
Job	Software Developer, Barista
Tool	Python, conga for mailmerge, risk assessments
Course	MSc Business Innovation with International Technology Management, Foundation Course T102 in Technology
Interest	cycling, travelling the canal network, Les Mills Bodypump

Table 1.1: Example information snippets for each topic

Intek search queries multiple data sources using multiple extraction techniques, with the aim of delivering multiple "factoids" each containing a different type of information concerning the snippet that has been searched for. The interviewer may use these factoids as a source of information for unexpected tests of expected knowledge.

The Intek UI is designed to be as simple and straightforward as possible. The aim is that novice interviewers should be able to follow the layout intuitively during an interview, whilst multi-tasking, with no issues. The layout of factoids and topics is based on the iterative CCE structure.

Figure 1.3 shows the Intek UI with multiple topics loaded, with some factoids opened and information highlighted for use in an interview.

Intek is designed to support any interviewing scenario in which deception detection might be a useful element. For this thesis Intek has been tailored toward job interviewing by including topics that might appear on a CV and supporting the preparation of an interview "script" in advance from a CV as well as supporting search during an interview in response to new information.

We fully discuss Intek's concepts, requirements, design, development and evaluation in Chapter 3.

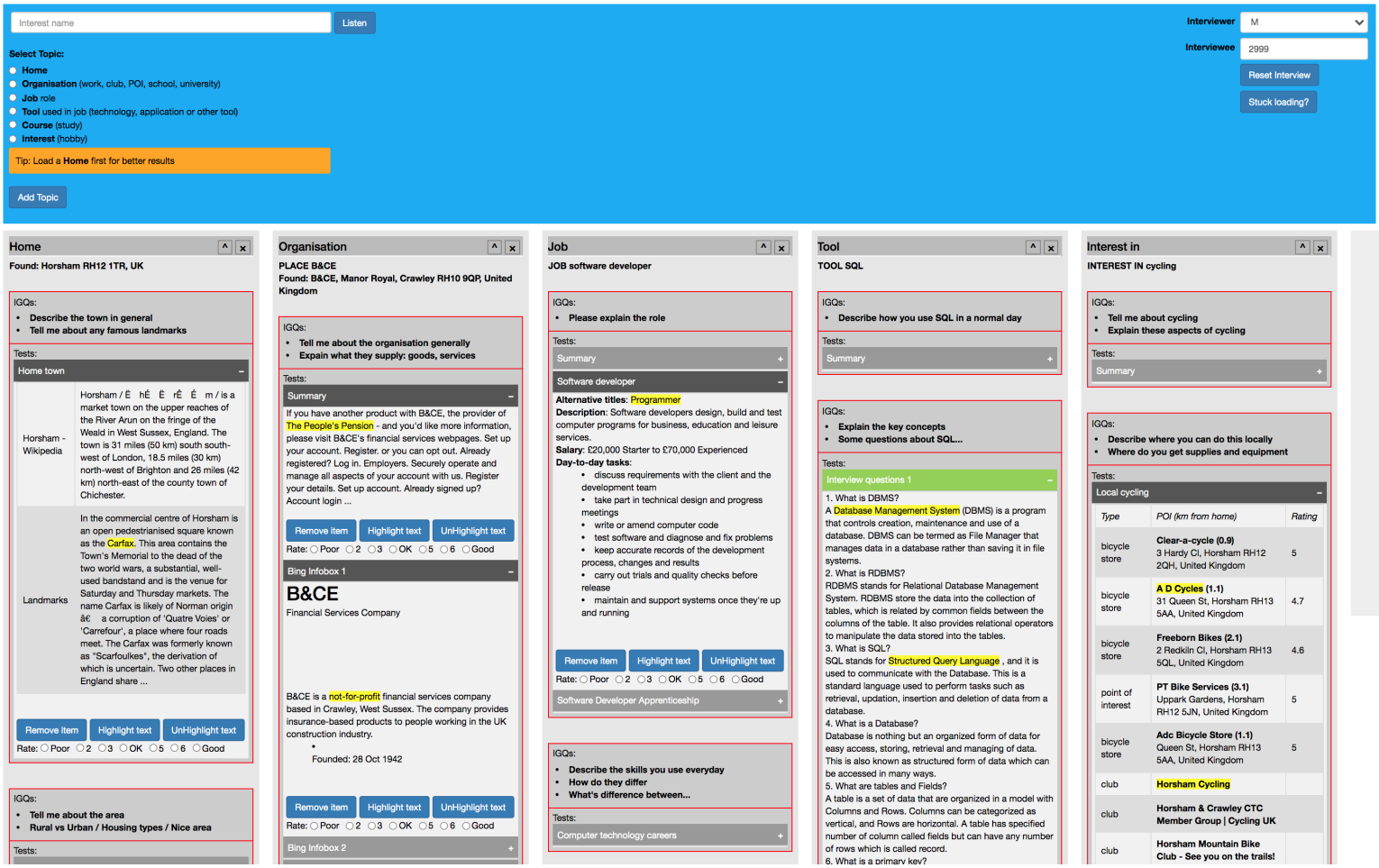

Figure 1.3: The Intek User Interface with five topics loaded, with some test questions highlighted.

1.6 OPPORTUNITY IN WEB NER

In this section we give some key definitions before describing the motivation and details of our work supporting an element of the Intek back-end with a novel web NER approach.

Named entity recognition (NER) is the identification of the proper names of objects in text, for example *John Smith*, *Apple Inc.* or *Brighton*. Natural language processing (NLP) is a research area concerned with the extraction of information from natural language, usually text-based sentences. Information extraction (IE) is a research area concerned with the extraction of information from unstructured or semi-structured sources, such as the web.

While most of the underlying NLP technology we used to create Intek was "off the shelf", we discovered an opportunity to investigate a novel approach to web NER leveraging HTML tags. Chapter 2

contextualises this work in the literature, while Chapter 4 details our work in this area. We now briefly summarise our work.

While creating the Intek extraction routines, we had difficulty extracting entity names from web sites in some circumstances, especially from very short or HTML rich "sentences"/spans, for example headings contain a single name. On investigation, we found that web extraction techniques (wrappers) are well suited to dealing with HTML markup, but not suitable for natural language sentence-based NER, whereas NLP NER does very well with sentence-based extraction, but does not deal with HTML markup at all (generally HTML is discarded).

We investigated whether HTML tags might improve NER performance by comparing Text+Tags sentences with their Text-Only equivalents, over varying datasets, free-text segmentation methods and NER models. We used a simplified tags as tokens approach to sentence processing, which required minimal extra pre-processing and between 3% and 11% extra processing time. We observed F1 improvements of between 0.9% and 13.2% for Text+Tags over all datasets, variants and models. These datasets, variants and models had quite different characteristics, proving the flexibility and adaptability of our novel approach.

The contributions of this work are listed in Section 1.8. We now return to the evaluation of the Intek application.

1.7 EVALUATION OF INTEK IN A JOB INTERVIEWING STUDY

In order to evaluate the Intek application, a study was undertaken to measure deception detection in job interviewing. 13 novice interviewers interviewed 111 interviewees over three conditions: "standard" interviewing; CCE interviewing; CCE+Intek interviewing. The Standard condition was based on Chartered Institute of Personnel and Development techniques and was designed to emulate a normal job interview that might take place in industry. The CCE condition brought in the CCE framework and techniques such as behaviour baselining, iterative test questioning with observation against baseline. Interviewers were encouraged to briefly research areas of questioning beforehand based on a supplied CV. Lastly, the CCE+Intek condition was identical to CCE but with interviewers using the Intek application for preparation and interviewing. This study is described in more detail in Chapter 5.

Condition		Actual TT	Actual D
Std	Predicted TT	9 (64.3%)	13 (65.0%)
Std	Predicted D	5 (35.7%)	7 (35.0%)
CCE	Predicted TT	19 (73.1%)	12 (66.7%)
CCE	Predicted D	7 (26.9%)	6 (33.3%)
Intek	Predicted TT	**13 (76.5%)**	**5 (31.3%)**
Intek	Predicted D	**4 (23.5%)**	**11 (68.8%)**

Table 1.2: Study results for the interviewer overall deception judgement of truth-teller (TT) or deceiver (D).

Table 1.2 shows high-level results for Intek that represent a two-fold increase in deception detection performance over our baseline approaches. Intek gives accuracy in the top 1% of studies in Bond Jr and DePaulo (2006). Furthermore, this result was attained in a realistic job interviewing scenario, using novice interviewers without specialised contextual knowledge in the topics discussed. The CCE method appears only to perform well in a job interviewing scenario with Intek support.

1.8 CONTRIBUTIONS AND THESIS STRUCTURE

The contribution of the work presented in this thesis may be usefully split into the communities that might benefit, as listed below. We explain these contributions more fully in Chapter 6 Section 6.2.

We have shown that deception detection can be greatly improved, primarily by asking better test questions. Intek allowed us to investigate the effects of different types of question and the subject knowledge these questions conferred on the interview process at a granular level. This investigation lends support to certain theoretical work in deception detection. We have also identified additional techniques, for example linguistic analysis, that might be integrated into our user interface to further enhance detection deception. These insights may be of interest to those researching deception detection.

We have developed a successful practical interviewing tool. Our development process and the data we have gathered has given us insight into design implications such as the most effective topics, data sources, extraction and presentation techniques to use in developing an interviewer support tool, given the constraints. We have also considered how Intek might be applied to other interviewing scenarios, such as security interviewing, automated online interviewing or in-

tegration with other aspects of industry standard job interviewing. These insights may be of interest to researchers or industry developing these tools.

We have gathered substantial data in application logs, expert ratings, questionnaires, videos and transcripts that might be of use to both of the above parties.

We present a flexible method for web NER that leverages HTML to improve NER performance. We have gained insight into how this method might be expanded to use a deeper representation of HTML or visual web page structure. Also, our use of HTML might be extended to other web-related NLP tasks, such as language model construction, knowledge discovery or information retrieval. These insights may be of interest to the information extraction or natural language processing communities.

Our method for web NER enables those that aim to extract typed entities from the web at scale to use a single approach for record-based and text-based extraction. This approach may be of use to industry, especially Competitive Intelligence or Business Intelligence practitioners.

This thesis is structured into the following chapters:

- In Chapter 2 we contextualise our work by reviewing research in the areas of interviewing and deception detection, technology for deception detection, the fields of information extaction and named entity recognition as a whole and specific areas of NER pertinent to our web NER approach.
- In Chapter 3 we give an overview of CCE and Intek concepts, then detail each step in the development lifecycle in two main areas: front-end UI and back-end information extraction.
- In Chapter 4 we detail our approach for the extraction of entities from web free-text.
- In Chapter 5 we describe the key elements of our empirical study in which we evaluate Intek. We then present our results and discussion in an order that builds support for Intek's contribution to the essential factors in interviewers' ability to detect deception. We also give details of a usability evaulation of Intek.
- In Chapter 6 we summarise this thesis, give more detail on our contributions to knowledge and discuss limitations of our research and future work.

2

RELATED WORK

The related work in this chapter is split into five sections. Firstly, we review interviewing frameworks to give insight into the typical sequence of events for investigative interviewing. We then examine approaches for detecting deception and the different factors that might be used. We then look at technology that has implemented some of these techniques. We then review research across the whole of information extraction and named entity recognition to give background to the competing areas which our web NER work unifies. Finally, we review approaches specific to our web NER work.

2.1 INTERVIEWING FOR INFORMATION GATHERING

This section contains methods for eliciting information from witnesses or suspects. Some of these methods include testing and challenge of accounts, but their primary purpose is building a picture of events. The general interview structure of CCE and thereby Intek is based on some of these methods.

Conversation management (CM) (Shepherd, 1988; Shepard, 1993) is an interviewing framework for building a detailed, but accurate, picture of relevant events. It is based on developing a relationship of respect between interviewer and interviewee that can be used to deal with uncooperative interviewees. CM begins with rapport building, based on principles of RESPONSE (Respect, Empathy, Supportiveness, Positiveness, Openness, Non-Judgemental Attitude, Straightforward talk, Equals talking across to each other). CM also has a specific method for recording details in sequence that supports the gradual build up and review of information in an interview.

The cognitive interview (CI) (Fisher and Geiselman, 1992) emerged around the same time as CM to support retrieval of information, especially from cooperative witnesses. CI uses four techniques that can be used to aid recall: mental reinstatement prompts the witness to recreate the context of the event through details such as sights, sounds and emotions; in-depth reporting aims at reporting every detail, even those that appear insignificant; recalling events in different order, par-

ticularly from end to beginning; Report from the perspective of other involved parties.

The PEACE framework (Green et al., 2008) was adopted by the British police in 1992 in response to deficiencies with existing methods. PEACE is aimed primarily at gathering relevant reliable information as completely as possible. PEACE is an acronym for the process: Planning through familiarisation with the case and any evidence; Explain and Engage by describing the outline of the interview and building rapport with the interviewee; The Account stage employs CM or CI to elicit details and accurate information, with additional clarification and challenge if necessary; Close by summarising the interview; Evaluate through analysis of the interview and information gained to identify further enquiries and improvement needs.

The enhanced cognitive interview (ECI) (Köhnken et al., 1999) was developed to address issues with the original Cognitive Interview. Specifically dealing with anxious, inacticulate or unsure witnesses and to lend a consistent overall interviewing strategy. ECI uses the same four techniques as CI to motivate recall, while adding rapport building, allowing the interviewee to talk freely and appropriate use of effective question types and non-verbal behaviours.

CCE was influenced by all these techniques, particularly the cognitive interview and the PEACE framework, with which it shares some stages.

The Reid technique (Inbau et al., 2013) is included as an extremely popular interviewing technique, especially in United States police forces. It consists of three stages: a case analysis stage in which known and unknown details are discussed and the interview planned; a non-accusatory interview for information gathering, but also to apply behavioural indicators to establish truth or deceit; if signs of deceit are present, then a third interrogation stage is carried out. The interrogation stage has received criticism for being confrontational, coercive and guilt-based with a higher chance of contamination and misclassification.

2.2 FACTORS IN DECEPTION DECTECTION

This section examines methods that deal specifically with deception detection. These methods may be broken into the three main types of factor used to discriminate truth-tellers from deceivers: nonverbal behavioural cues including gestures, movements and expressions; verbal

indicators that include various types of analysis of the words spoken in an interview; contextual veracity which relies on some degree of expert knowledge to elicit and check for contextual contradictions and implausbility in the topic of discussion.

2.2.1 *Nonverbal Behavioural Cues*

Ekman and Friesen (1969) noted that nonverbal behaviour is not so easily controlled by a deceiver as verbal and may evade self-censoring or "leak", giving clues to deception. Ekman explored the possibility that the combination of neurological and cultural factors between both parties in an interview interaction might produce specific body movements and facial expressions which escape efforts to decieve and which offer indications of deception.

Zuckerman et al. (1981) identified four factors involved in deception that influence behaviour that may be detectable. Firstly, deceivers' attempts to control and mask their deceptive behaviour may appear rehearsed and lacking spontaneity. Deceivers may try too hard offering excessive information. Also, since they may mask some behavioural aspects better than others, inconsistencies may arise, for example between face and voice. Secondly, deception may produce autonomic arousal which has a variety of possible causes and might be measured through behaviours such as pupil dilation and blink counts. Thirdly, perhaps related to arousal, deception is also linked to specific emotions, most commonly guilt about deceiving and anxiety of being caught, also joy in having deceived. Joy after deception or "duper's delight" as it is also known, was noticed in video recordings from our study in this thesis. Lastly, due to the difficulty of creating lies that are internally and externally consistent, gestures or speech characteristics indicative of complexity, thinking time or pauses may arise.

Interpersonal deception theory (IDT) (Buller and Burgoon, 1996) aims to explain how interviewees deal with deception on a conscious and subconscious level during the process of an interview. IDT found "leakage" due to emotion is manifested most overtly in nonverbal behaviour, some studies indicating over 90% of emotion is communicated non-verbally. Emotional content leakage might manifest in micro-gestures, such as lip corner pulling and cheek raising with happiness or brow lowering and lip stretching with unhapppiness.

Leakage might also manifest as general facial expressions of emotion, gaze, gesture or touch.

DePaulo et al. (2003) performed a meta-analysis of 1,338 estimates of 158 cues to deception to gain insight about the discriminatory value of behaviour in deception detection. They found that in general, deceivers are less forthcoming in their accounts than truth-tellers and those accounts are less compelling, containing fewer imperfections and unusual items. Deceivers also make a more negative impression and tend to be less relaxed. However, they found many behaviours were weakly or not at all linked to deception.

In aviation security, the most widely used security procedure "suspicious signs" (Reddick, 2004) is based on the identification of behavioural indicators in response to scripted security-related questions. The indicators used include verbal and nonverbal behaviour, including disposition and appearance.

Vrij and Granhag (2012) offer guidelines for more effective questioning in order to maximise behavioural cues. They encourage researchers to focus more on the quality of elicitation, rather than quality of detection. They recognise that lying is cognitively more difficult than telling the truth. They aim to build on this cognitive load through the use of techniques such as reverse ordering, unanticipated questions, information gathering with open questions and the use of a model statement. A model statement gives deceivers an example of the level of detail required from them. They recommend the strategic use of evidence (SUE) technique which we discuss in the next section.

2.2.2 *Contextual Veracity*

Contextual veracity moves away from observing cues of behaviour toward checking accounts against expert knowledge and evidence. This move was driven partly by a survey carried out by Park et al. (2002) which indicated that interviewers recall successfully detecting deception through two main techniques: comparing against some type of evidence or an honest confession. Behavioural cues were rarely indicated as helpful. Levine (2015) reports that behavioural cues perform only slightly better than chance and that improved accuracy is seen across a variety of methods using contextualised communication, including veracity checking against expert knowledge and use of evidence. Deceivers can also be persuaded to confess using these techniques.

One of the first techniques to show better than chance performance in deception detection was the strategic use of evidence (SUE) (Hartwig et al., 2006). Using SUE, evidence is withheld until an interviewee contradicts this evidence during contextual discussion. The evidence is then gradually revealed and the interviewee asked to explain the inconsistencies. The initial contradiction is a good indication of deception, but the gradual revelation of evidence can put the deceiver in a very difficult position, proving the deception or eliciting a confession. Conversely, a truth-teller's account is likely to be consistent with the evidence and, if not, they are likely to be forthcoming in explanation. SUE has been shown to perform well in lying adults (Hartwig et al., 2011) and children (Clemens et al., 2010), also when suspects lie about past actions (Hartwig et al., 2005) and future intentions (Clemens et al., 2011). SUE is not necessarily relevant to job or security interviewing where evidence does not exist per se, but evidence can be treated as an extreme form of contextual expert knowledge.

The next step from use of evidence, is to compare interviewee dialogue against facts based on contextual expertise. This approach is useful where evidence is not available. Blair et al. (2010) suggest the interviewer should be a judge "situated in a meaningful context". This gives the interviewer the opportunity to detect deception directly through contradictions against subject facts as well as assessing the plausibility of statements given expert knowledge of how the subject operates in the real world. Blair et al. (2010) found deception detection accuracy improved when judges where trained in contextual background knowledge. Similar increases in accuracy were attained when using interviewers already familiar with the topic under discussion (Levine and McCornack, 2001; Reinhard et al., 2011, 2012). Levine et al. (2014a,b) found that interviewers expert in active questioning to elicit diagnostic information as well as contextually expert, performed very well. Additionally, students watching recordings of these interviews also performed very well, indicating that expert questioning for diagnostic answers was more significant than expertise in the signs of deception.

The studies in this section have produced some impressive results using contextual and diagnostic questioning expertise. However, they are not directly comparable to CCE or the study presented in Chapter 5 of this thesis. The main reason for this is that we do not use evidence or experts, mainly because in general interviewing the subjects

of discussion are heterogenous. The interviewees used in our study are novice interviewers and are also unlikely to be expert in many of the items discussed. However, CCE does approximate diagnostic questioning through the use of questions that are veracity testable *in theory* and Intek builds upon this through the use of topic familiarisation to build some level of expertise.

CCE could be seen as a behavioural approach, as it relies entirely on the observable behaviour of the interviewee to discern deception. However, CCE uses the practical method of behavioural baselining to gain a personalised insight into "normal" behaviour with which to compare further behavioural responses to questioning. This method has proved more effective than the one-size-fits-all behavioural indicators discussed in Section 2.2.1. Additionally, CCE's probing test questions that are veracity testable *in theory* (in practice interviewers don't know the answers), are a kind of diagnostic questioning without knowledge. For this reason CCE might be described as a hybrid of the methods discussed so far. CCE's main elements are discussed further in Chapter 3 Section 3.1.

2.2.3 *Verbal Indicators*

Interviewee statements or interview transcripts can be analysed using various methods and verbal criteria. These methods are of interest as background to deception detection and as possible future complimentary techniques for Intek.

Statement validity analysis (SVA) (Köhnken, 2004) suggests there is a detectable difference between products of experience (truths) and products of imagination (lies). Interviewee statements are reviewed using criteria-based content analysis (CBCA) which uses nineteen criteria that are thought to be more present in truthful accounts than deceptive, for example self-corrections "oh, no, sorry". The CBCA is then itself reviewed for validity by considering additional influences on the result, such as interview quality. SVA is a robust, yet time-consuming approach, that considers the whole case as well as the written statement.

Reality monitoring (RM) (Johnson and Raye, 1981; Sporer, 2004; Masip et al., 2005) suggests memories externally-derived from perception of events (truths) have a different quality to those internally-derived via thought and imagination (lies). Truths typically include more contextual details (e.g. space and time), perceptual details (e.g.

shapes and colours) and semantic information. Lies typically include more cognitive operations at around the time of encoding. Discriminating the initial source of a memory by analysing these types of language is known as reality monitoring, which clearly has its uses in deception detection.

Scientific content analysis (SCAN) (Sapir, 1987) is a well-used tool in investigative interviewing. SCAN is similar to SVA in that a detailed interviewee statement of all activities is analysed using list of criteria such as: increased use of pronouns (truth); missing information (lie). Nahari et al. (2011) compared SCAN with RM, finding RM discriminated significantly between truth-tellers and liars, whereas SCAN did not, bringing the use of SCAN into question.

The verifiability approach (VA) (Nahari et al., 2014a) is similar to contextual veracity methods in that it requires the elicitation of verifiable details in a statement from an interviewee. This statement is then examined using the theory that deceivers will report carefully, avoiding giving checkable details where possible. The count of verifiable details and the ratio of verifiable to non-verifiable details can both be used to discriminate truth-tellers from liars. Nahari et al. (2014b) found VA to be robust against "countermeasures" by informing half of their participants that statements would be checked, which increased deception detection by encouraging truth-tellers to write more.

We now move on to discuss actual technological implementations of techniques similar to those in this section.

2.3 TECHNOLOGY FOR DECEPTION DETECTION

This section examines implemented technology related to interviewing for deception detection. Firstly, we look at technology for the detection of behavioural and verbal indicators of deception and veracity checking. Secondly, we examine technology for the creation of test questions.

2.3.1 *Detecting Deceptive Factors*

The main focus of technology support for deception detection in interviewing has been identifying deceptive behaviour after the interview, from videos or transcripts.

Enos (2009) was one of the first to consider high-quality speech recordings of interviews from their own CSC interview corpus, using

acoustic, lexical and subject-related features while comparing various machine learning algorithms. They observed performance better than chance and much better than human listeners on a similar task.

Levitan et al. (2018) used linguistic features with the addition of follow-up question counts on the CSC corpus, comparing machine learning algorithms to achieve a F1 score of 72.72%. Various levels of linguistic features have been used from word frequencies, syntax and semantics. Linguistic feature-based approaches have been used in other similar areas, such as political fake news detection (Wang, 2017; Rashkin et al., 2017).

O'Shea et al. (2018) trained an artificial neural network to recognise 1-second micro-gestures expressed by the faces of participants in security interview recordings, to achieve an accuracy of 73.66%. A micro-gesture is a non-verbal behaviour such as "face movement angle-change". Crockett et al. (2020) uses the same approach to diffentiate male and female gestures.

Another approach to deception detection is veracity checking claims made in an interview. The fact-checking process is made up of four tasks: estimating the check worthiness of a claim; retreiving relevant previously fact-checked claims; retrieving evidence for new claims; verifying claims using this evidence. To our knowledge, only one system, Claimbuster (Hassan et al., 2017), attempts all these tasks in an end-to-end approach that could be applied to textual interview dialogue. Claimbuster uses various machine learning models and linguistic features to identify important claims, it then tries to match claims with a database of facts pre-checked by humans. For new claims, Claimbuster combines results from Wolfram Alpha, Google search question answer and top website results to attempt a verdict if clear discrepancies exist, otherwise evidence is reported to the user for evaluation. The Claimbuster pipeline is applicable to some of the claims that might be expected in a deception detection interview, for example competence oriented questions or location-based information. However, Claimbuster is not designed to check personal claims at the episodic level.

Also notable in the area of fact checking is the fact extraction and verification (FEVER) dataset (Thorne et al., 2018). The FEVER dataset was created to drive research in the extraction of fact-supporting evidence and fact verfication or rejection. FEVER consists of 185,445 claims extracted from Wikipedia, which have been manually labelled

as supported, rejected or lacking evidence. If claims are supported or rejected, supporting evidence is appended to a claim.

Following Vrij and Granhag (2012), Intek's approach does not focus on detecting the signs of deception, but rather on supporting the interviewer to generate excellent test questions in advance. However, the systems discussed above may be complimentary to our approach and might be fruitfully combined in future work, particularly if they could be adapted to operate closer to real-time, rather than analysing interview recordings.

2.3.2 *Question Generation*

Question generation for deception detection interviewing ideally requires unexpected tests of expected episodic knowledge. For episodic knowledge to be tested, automatically generated questions should be as specific as possible to an individual interviewee's experience. For this to be possible, questions should be generated dynamically to fit the available interviewee information, as it is gathered.

The many resources available on the web containing pre-defined static questions, based on broad competency (TestGorilla, 2021) or psychometric (SELECTPro, 2021) categories, are of limited use for episodic deception detection.

Automatic question generation (AQG) systems (Revision.ai, 2021; Zeng and Nakano, 2020) extract syntactically correct questions from text or knowledge base sources. AQG is frequently used to generate alternative answers, known as "distractors", to support multi-choice questioning given a specific subject. AQG operates on information sources containing facts that have already been extracted, it is not concerned with generating dynamic subject matter for deception detection questioning.

Natural lanaguage generation (NLG) systems have been used to provide dynamic follow-up questions in response to interviewee dialogue using various approaches: keyword substitution into pre-defined templated answers (Inoue et al., 2021); reduced sentence patterns mapped to follow-up responses using seq2seq (Su et al., 2018); keyword mapping to ConceptNet triples for substitution into templated answers (Su et al., 2019); the use of large pre-trained language models to direct generate follow-ups (SB et al., 2020). Follow-up question generation is largely aimed at automated interviewing scenarios; its flexibility is limited as topics, initial questions and templates are scripted

and therefore predictable. The questions generated are conversational and general in nature, not applicable for episodic questioning.

To our best knowledge, only one system (Tsunomori et al., 2015) directly addresses question generation for deception detection interviewing. The focus of this system is the analysis of the impact of questioning on deceptive behaviour, rather than a usable interviewer support tool. They model questions used in recorded interviews using hidden Markov models based on fine-grained question categories. They then deploy test questions, using the appropriate model, only if the deception detection evaluation module gives the previous interviewee response above 50% chance of being a lie. The deception detection module is trained on a Japanese dataset of recorded video and evaluated using the same acoustic, linguistic and subject-based features as Enos (2009). They find that these generated follow-up questions slightly increase deception detection. However, the questions generated are similar to those from NLG systems; conversational, general and somewhat basic, "Is that true?" for example.

Computer assisted investigative reporting (CAIR) has a similar aim to Intek, in that reporters or investigators seek to extract relevant information by analysis and cross-reference from a plethora of data (Garrison, 2020). CAIR uses tools such as search engines to locate suitable data, then generic spreadsheets or databases to analyse and cross-reference. More recently tools that combine search and analysis, such as LexisNexis (2021), have been used. The effectiveness of CAIR is hard to quantify as results are not conclusive and will differ on a case-by-case basis. Intek might be seen as a more constrained and focussed, automated form of CAIR.

2.4 INFORMATION EXTRACTION AND NAMED ENTITY RECOGNITION BACKGROUND

In this section we review techniques for using HTML tags and visual features in information extraction, we then examine key NER systems and finally methods for generating labelled training data for NER. This section provides background for specific literature in Section 2.5 that relates to our work in Chapter 4.

2.4.1 *Leveraging HTML and Visual Features*

Web content mining (WCM) is the subset of web data mining concerned with the extraction of information from repetitive data structures in the content of web pages. WCM typically uses HTML-aware "wrappers", small programs containing extraction rules for a given semi-structured pattern or template. Wrappers are hand-crafted and rule-based in earlier techniques and more recently automatically induced using variants of DOM tree or string alignment. WCM tends toward a business focus, using techniques with high understandability and maintainability, consumed by downstream tasks such as Business Intelligence or Competitive Intelligence. WCM typically deals with semi-structured content such as tables or lists where HTML tags are essential delimiters. Unstructured text data is left for plain-text extraction and processing using NLP techniques.

Manual wrapper creation is typically based on rules and constraints, using tools such as XQuery, XLST and regular expressions. It is labour intensive and the wrappers created are brittle, but viable in simple single-site cases. Wrapper programming languages, such as WDEL (Li and Ng, 2004), reduce some of the effort of wrapper creation.

Automatic wrapper induction is performed using labelled template examples or in automatic extraction using seed page(s) to compare, discover and extract wrappers from repetitive structures. This is seen in tools such as RoadRunner (Crescenzi et al., 2001) which compare a set of web pages, analysing similarities and differences in their flattened DOM trees, creating a set of regular expressions in which disjunctions contain useful content to be extracted. IEPAD (Chang and Lui, 2001) uses iterative pattern mining and tree alignment to discover extraction rules from web pages supplied by a user. Zhai and Liu (2006) find record structures and fields by DOM sub-tree alignment using edit distance. Matches are filtered using visual cues then aligned with knowledge base slots. Qiujun (2010) extracts news content by comparing against similar pages on a different web site published in the same topic near the same date. The comparison disjoint is used to extract useful content. Strings are matched using Edit Distance. WebSets (Dalvi et al., 2012) uses unsupervised clustering of similar entities in HTML tables which are then assigned types using Hearst patterns. Dexter (Qiu et al., 2015) detects product specifications from HTML tables and lists using an unsupervised two stage process that compares similar structures for a product, then

filters the wrappers generated with noisy annotators. Gogar et al. (2016) construct and filter DOM trees by tree alignment using normalised HTML tags as nodes, then use a deep neural network based on visual and textual features to learn wrappers automatically. TANGO (Jiménez and Corchuelo, 2016) induces wrappers on the web using HTML, CSS and user-defined features which are mapped to knowledge base slots using a catalogue. BigGrams (Mirończuk, 2018) uses a semi-supervised technique based on user-identified seeds organised into a taxonomy. These seeds are used to extract wrappers that use HTML tags to delimit slot-based information. They note that HTML simplification is an important performance factor. These approaches to wrapper induction typically focus on semi-structured page elements such as lists and tables where tags are most fruitful, rather than free-text sentence content.

Sleiman and Corchuelo (2014) bridge the gap between wrapper generation and NLP techniques using finite state transducers. Transducers are learned from a training set of web documents in which "slots" of various types have been labelled, for example Book, title, author. Transducers represent the structure and sequences of records within the page and fields within records, while neural networks are used to model the transitions between these records and fields, including both words and HTML tags. We build on this approach in Chapter 4 using state-of-the-art NER techniques to process HTML-containing text sequences.

Large information extraction systems offer a performant and accessible approach to information extraction. Previously based on rules, they now typically use a combination of techniques including machine learning. Examples are GATE (Cunningham and Bontcheva, 2011), DeepDive (Niu et al., 2012), DEiXTo (Kokkoras et al., 2013) and SystemT (Chiticariu et al., 2018).

This section examined approaches designed to extract information from HTML structures. These techniques do not address the information contained in natural language free-text sequences, even if those sequences contain HTML.

2.4.2 *State-of-the-Art Named Entity Recognition*

This section examines NLP approaches to the extraction of entities from free-text. These techniques remove any HTML that might be present before processing. F-measure NER scores presented in the

format F1 xx.xx are based on CoNLL-2003 English NER dataset evaulation, unless otherwise stated.

NER rule-based systems have evolved from entirely hand-crafted rules and patterns, for example NetOwl (Krupka and Hausman, 1998), to high-level rule-based languages like NERL (Chiticariu et al., 2010), to complete systems incorporating rules like GATE/ANNIE (Cunningham and Bontcheva, 2011) and SystemT (Chiticariu et al., 2018). Rule-based systems can combine high-precision with good understandability and maintainability. Most of these systems have been updated to use an element of machine learning for increased performance.

Initial machine learning approaches used local/shallow, global or externally-linked features, discovered and engineered by researchers. While these hand-engineered features performed well, they may not have fully aligned with and maximised the latent features in the data. Task and language-specific features such as gazetteers or part-of-speech can degrade generalisability. Examples of feature-engineered machine learning systems are: Li et al. (2005) who pre-process using ANNIE to provide syntactic, semantic, part-of-speech and named entity type features, which are then trained on using an uneven margin support vector machine (SVM) and perceptron classifiers achieving F1 88.3; Ando et al. (2005) augment a labelled dataset with token-based, part-of-speech and token-context features from an unlabelled set to achieve F1 89.31; Finkel and Manning (2009) use a probability distribution of task feature parameters over multiple domains to influence domain-specific parameters of a conditional random field (CRF) model achieving F1 85.81; Passos et al. (2014) use a CRF on top of skip-gram word-embeddings (Mikolov et al., 2013) clustered around named entity types. This is achieved by injecting named entity type-specific curated lexicons extracted from US census data and Wikipedia into the learning process, achieving F1 90.90; Agerri and Rigau (2016) use shallow token and sentence level features with a weighted combination of Brown, Clark and Mikolov skip-gram clusters generated from Reuters, Wikipedia and Gigaword datasets using a perceptron model achieving F1 91.36.

Feature-inferring neural network systems deliver benefits of reduced feature engineering time and better generalisability, given a good generalisable dataset, that have led to their recent popularity. They can be grouped generally into systems that use "classic" shallow one-sense-

per-word embeddings and deep pre-trained language model based systems.

Shallow word-embedding systems are separated by those that process sequences at the word-level, the character-level or a combination of the two. Word-level approaches such as Collobert et al. (2011) were the first to use a convolutional neural network (CNN) model on a large unlabelled dataset to generate a general shallow language model, LM2. LM2 was then applied across multiple NLP tasks, with output passed to a CRF for sequence labelling NER, scoring F1 81.47. Huang et al. (2015) used a bi-directional long short-term memory (Bi-LSTM) model outputting to a final CRF layer for prediction with gazetteers and SENNA embeddings, achieving F1 90.10. Character-level approaches, such as Gillick et al. (2015) using a character sequence to sequence LSTM transformer model achieving F1 86.50. Combined word and character approaches are: Chiu and Nichols (2016) represent a word by its embedding plus a convolution over its characters, input into a Bi-LSTM, using additional capitalization and gazetteer features, to achieve F1 91.62; Ma and Hovy (2016) layer a CRF over the Bi-LSTM for prediction and using no extra features achieve F1 91.21; Yadav et al. (2018) add a separate representation of word prefixes and suffixes to allow the model to better predict over unseen words achieving F1 90.86.

Recent hardware has made deep language model (LM) training possible on massive datasets. These models have been used to great effect in many NLP tasks. A LM is essentially a deeper multi-layer version of the shallow word-embedding model, the extra layers allowing higher-level representations, such as word-sense per sentence to be captured. LMs typically use unsupervised pre-training tasks, such as the masked language model, which allows any text corpus to be used for pre-training, providing massive amounts of data. Using LMs as a starting point for other tasks allows domain-specific tuning with relatively small datasets. ELMo (Peters et al., 2018) trains a deep Bi-LSTM model using a predict next word task on the 1 billion word benchmark dataset. ELMo combines all Bi-LSTM hidden layers to generate a final embedding. ELMo embeddings are used as additional word-level features in an existing model. Using a character-based CNN with stacked Bi-LSTM and CRF, ELMo achieved F1 92.22. GPT (Radford et al., 2018) trains a multi-layer unidirectional transformer decoder using the predict next word task on the 800 million word multi-genre BooksCorpus dataset. The transformer model uses

12 layers of combined self-attention, identifying the most semantically important words. GPT-2 (Radford et al., 2019) scaled GPT up to 1.5 billion parameters, more than 10 times that of GPT. GPT-2 was trained on text from 8 million websites with a scaled down "small" version at 117 million parameters on same dataset. GPT LMs are used by fine-tuning the whole LM network on the downstream task. BERT (Devlin et al., 2018b) trains a bidirectional transformer model on the BooksCorpus plus English Wikipedia containing 1.5 billion words. Two models are generated, Base using 12 layers (for comparison to GPT) and Large using 24 layers. BERT uses a masked language model task rather than predict next word, masking 15% of words at random, with the objective of predicting these masked words. This task considers context from the whole input sequence and as such is inherently bi-directional. With fine-tuning BERT-Large achieves F1 92.8 and BERT-Base F1 92.4. CSE (Akbik et al., 2018) uses a character-level language model trained on a predict next character task on the 1-billion word corpus (Chelba et al., 2013). This task is unaware of words and sentence boundaries, generating contextual embeddings that result in a smaller vocabulary, allowing more efficient training and better character level features. Using a standard Bi-LSTM/CRF combination this system achieves F1 91.97 and when combined with pre-trained GloVe embeddings and NER task-specific fine-tuning achieves F1 93.09.

This section identified state-of-the-art approaches to NER on free-text sentences. These techniques are compatible with the inclusion of HTML, as long as HTML is introduced as tokens in a sentence sequence. In our experiments in Chapter 4, we select a Bi-LSTM system with character-level CNN as in Chiu and Nichols (2016) for its training speed, high-performance and potential ability to discover latent features in HTML structure. We also fine-tune a pre-trained BERT LM (Devlin et al., 2018b) model on our NER task for comparison. BERT is selected for ease and speed of fine-tuning with the potential benefits of the deeper network extracting higher-order features from HTML. We now move on to describing approaches to training data generation for machine learning models.

2.4.3 *Generating Training Data*

The original and typically most accurate method of acquiring labelled data for NER machine learning training is hand-labelling examples. This manual approach is a time-intensive process that has limited

hand-labelled datasets to a specific task and/or small size. User interface support through tools such as Scrapinghub's Webstruct or LightTag might be used to increase the usability of the labelling experience. While active learning techniques can be used to increase efficiency by guiding the labelling process to the most informative examples. Shen et al. (2004) guide learning using measures to quantify informativeness, representativeness and diversity. Shen et al. (2017) show that a variety of active deep learning techniques can give nearly state-of-the-art results using a much smaller dataset with concomitant fast training time. Tools such as Explosion's Prodigy combine UI support with active learning. Labelling is a very parallel task lending itself to crowdsourcing (Finin et al., 2010).

Semi-supervised and unsupervised approaches have the potential to generate large amounts of training data with minimal human intervention.

In the situation where large amounts of unlabelled data are available, semi-supervised techniques may be used to label some portion of this unlabelled data automatically. The main two semi-supervised techniques are bootstrapping and distant supervision, which share roughly equal popularity.

Bootstrapping may be used where no labelled examples are available. Starting from a small set of manually created seed examples, an iterative process is followed that: extracts rules from unlabelled data using the seeds; trains a classifier on those rules; extracts additional seeds using the classifier. Bootstrapping should choose initial seeds carefully to represent the desired feature space. The process can suffer from semantic drift in which each iteration drifts further from the initial design. Approaches differ in how the initial seeds are generated, either manually (Putthividhya and Hu, 2011), through domain adaptation (Jiang and Zhai, 2007; Wu et al., 2009), sourced from a knowledge base or gazetteer (Kozareva, 2006; Whitelaw et al., 2008) or extracted from patterns in text as in the BOA system (Gerber and Ngomo, 2012).

Distant supervision (DS) (Mintz et al., 2009) may be used where a substantial set of example entities exists in a knowledge base of some kind. This set of examples is aligned with a large unlabelled dataset using heuristics to perform labelling. The absence of iteration in distant supervision removes the problem of semantic drift. Distant supervision can generate large multi-domain labelled datasets, but is susceptable to producing noisy data due to ambiguous entity names

in the unlabelled data (producing false positives) or entities of interest missing from the knowledge base (producing false negatives).

Distant supervision has been used specifically for NER by Ritter et al. (2011) who use Freebase as a gazetteer source and to constrain a LabeledLDA topic model. Nothman et al. (2013) assign entity types to Wikipedia articles using article links to generate training data. Althobaiti et al. (2015) combine Arabic Wikipedia with bootstrapped ANERcorp data. In relation extraction, Mintz et al. (2009) coin the term "distant supervision" using Freebase as a source of 116 million relation instances which are matched to 1.2 million Wikipedia articles to perform labelling of the latter. Surdeanu et al. (2010) adapt DS to the TAC-KBP slot filling task, using Wikipedia infoboxes mapped to one or many slot types as a source of relations. These relations are then used to label the TAC-KBP official corpus and a full Wikipedia extract using extract string matching.

Despite the benefits of distant supervision, performance tends to be lower than supervised approaches due to the problem of noisy labelling. Various approaches have addressed the problem of noisy labelling. Surdeanu et al. (2012) apply a multi-instance multi-label (MIML) bag-of-instances approach. Sterckx et al. (2014) and Augenstein et al. (2016) filter ambiguous relations using a manually trained classifier and statistical ambiguity measures respectively. Lin et al. (2016) uses a CNN over unlabelled sentences to embed semantics in the unlabelled data. They then use selective attention between these semantic sentences and potentially matching relations from distant supervision. Only "really expressive" matches are selected for labelling.

Unsupervised approaches seek to draw inferences from unlabelled datasets without the need for human supervision and can therefore operate on very large datasets such as the web. The most common unsupervised techniques involve cluster analysis to find groupings and other patterns in data. Web information extraction techniques such as KnowItAll (Etzioni et al., 2005), TextRunner (Yates et al., 2007) and ReVerb (Etzioni et al., 2011; Fader et al., 2011) start with a few manually-created generic natural language seed patterns, for example "<noun phrase 1> such as <noun phrase list 2>" to extract a very large number of results from the web. These results may be uninformative and difficult to map to useful named entity or relation types. Some techniques similar to bootstrapping use an initial set of lexical rules, heuristics or examples to segment named entities by generating lists of exact

entities to match (Collins and Singer, 1999; Nadeau et al., 2006) or by including corpus statistics (Downey et al., 2007b). Techniques similar to distant supervision that use knowledge bases combined with contextual seeds as a topic model are Alfonseca and Manandhar (2002); Zhang and Elhadad (2013).

This section has identified techniques that could be used to generate training/fine-tuning data for the Bi-LSTM and BERT models selected for our experiments in Chapter 4. We use distant supervision to generate datasets for these experiments by matching DBpedia against web pages retrieved from search. Our work in Chapter 4 is a reaction to the literature in this chapter, as we find NER approaches that deal well with HTML and those that deal well with natural language text are mutually exclusive.

2.5 WEB NAMED ENTITY RECOGNITION

This section details literature that guided our decisions while implementing our approach to web NER in Chapter 4. We processed free-text from web pages into sentences including HTML tags as single tokens. We then evaluated the impact of this HTML by comparing the performance of the original free-text sentences with the HTML-infused versions. Two key decisions relate to what we define as a "free-text sentence" and how we represent HTML as tokens. These decisions are discussed in the two sections below.

2.5.1 *Segmenting Free-Text Sentences*

In order to extract free-text sentences from a web page, the level of granularity of the free-text containing sentence elements must be decided. The typical technique for NLP approaches is the indiscriminate removal of HTML from the whole-page (Bekkerman and McCallum, 2005; Downey et al., 2007a; Alfred et al., 2014; Singh et al., 2012), which can be easily accomplished using tools like NLTK (Bird et al., 2009) or BeautifulSoup (Richardson, 2007). Once the page text has been retrieved, any number of tools, such as Punkt (Kiss and Strunk, 2006), can be used to segment text sentences. Stripping HTML entirely can result in the inclusion of non-standard sentences, such as headings or any span with a concluding full stop. A more granular approach to sentence segmentation is to use specific HTML tags that are deemed free-text containing. Bunescu (2007) and Bunescu

and Mooney (2007) use BR, DD, P etc. as end of sentence indicators. Kohlschütter et al. (2010) uses heading tags (H1, H2, H3, H4, H5, H6), paragraph tags (P) and division tags (DIV) to encapsulate sentences. Kim (2017) uses paragraph tags only. Alternatively, all tags that contain text in the desired format can be extracted (Etzioni et al., 2005). In our web NER experiments, we compare a paragraph only and paragraph plus headings approach, both with additional sentence segmentation using Punkt.

2.5.2 *HTML Tag Representation*

We must also decide how to represent the HTML tags that appear in our segmented free-text sentences. Approaches for tag representation range from simplifying tags as much as possible, removing any attributes and style contained within the tag. Etzioni et al. (2005) use this approach in web NER to match HTML and text sequences using templates such as "<td>TOKEN vehicles</td>". Blohm (2011) uses a one tag per token approach for information extraction from Wikipedia. Mirończuk (2018) find simplified tags with attributes removed performed better than representing HTML sequences as separate characters. In an earlier approach to web information extraction, Soderland (1999) break down HTML tags by whitespace, for example "<font size = +2 >", is broken into five separate tokens. Freitag (1998) uses a feature engineering with machine learning approach, which defines many individual dimensions for text elements such as "in title", "in a", "in h1", noting a considerable performance increase using these features. Apostolova and Tomuro (2014) also use a feature engineering approach to represent visual aspects of text elements. They use features related to font size, colour and vertical position, as well as linguistic features for NER in presentation-rich online property flyers. Finally, Gogar et al. (2016) use a visual and textual approach to web information extraction, but use a convolutional neural network to extract features.

In our experiments in Chapter 4, we choose the simplest possible HTML tag representation of one token per tag, for example "<p>" or "<h1>". This allows us to focus only on the delimitation effects of tags on entities. Mirończuk (2018) show this to be a high performing approach.

3

INTEK: CREATING AN INTERVIEWER SUPPORT APPLICATION

This chapter describes the background and development of Intek, our interviewer support application. This chapter is structured as follows. We first give background information and examine the process and key terms of the controlled cognitive engagement (CCE) interviewing method, on which Intek is based. We then give an overview of Intek key terms and how these relate to CCE. We then describe the development life cycle used throughout this chapter and give details of several high-level design decisions. We then detail how the different steps in the development life cycle were applied for the Intek front-end user interface and for back-end information extraction.

3.1 INTERVIEWING WITH CONTROLLED COGNITIVE ENGAGEMENT

Intek is designed to support the controlled cognitive engagement (CCE) interviewing method for deception detection. We now describe the elements and process of CCE before showing how these elements are used in Intek. This section is broken down into sub-sections for the key elements in the CCE process, in the order they should be used: baselining; topic selection; information gathering questions; test questions; evaluation.

CCE is based on investigative interviewing techniques such as the PEACE model and the cognitive interview. CCE's motivation is to address the poor performance of behavioural cue methods overall and specifically the "suspicious signs" security screening technique used in airport security interviews.

Weakpoints are observed in the suspicious signs method: the lack of open questions and thereby lack of interviewee free-talk; the lack of baselining of "normal" interviewee behaviour for the interviewer to compare potential deceptive behaviour against; predictable questioning and lack of challenge of interviewee account.

CCE aims to remedy these issues by introducing initial behavour baselining and involving the interviewer as an active participant. Interviewers are asked to deliver unexpected tests of expected know-

ledge based on information supplied by interviewees during the interview. This probing questioning, using open question types to encourage further explanation, is designed to increase cognitive load in deceivers who must then expand their fabrications cohesively. The aim of this enhanced questioning is to elicit deceptive behaviour, which is detectable by its comparison to baseline.

These steps in the CCE process are shown in Figure 3.1 and explained in more detail below.

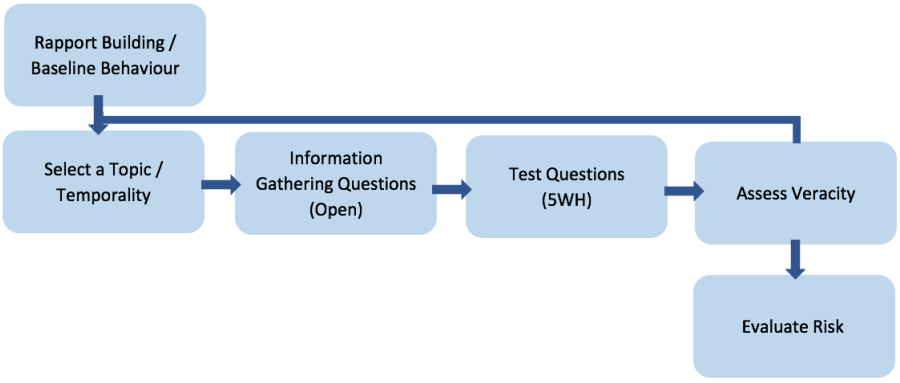

Figure 3.1: A high-level view of the iterative structure of a CCE interview.

3.1.1 *Baselining*

The baselining phase is easy to overlook or rush, but is critically important. It serves to build rapport between interviewer and interviewee, but more importantly, the interviewer has the opportunity to observe the interviewee's "normal" verbal and physical behaviour when challenged on a neutral topic. Neutral topics should be selected in which the interviewee is most likely to be truthful, an easy topic being the weather. More problematic topics are elements of the interviewee's appearance or their travel to the interview which might easily be regarded as a challenge. Good behavioural observation by the interviewer while asking questions is essential.

3.1.2 *Topic selection*

Identity/Intention	Topic	Aspects
Identity	Home	Family, Location, Type
Identity	History	Education, Work, Residence
Identity	Hobbies	Skills, Equipment, Places
Intention	Purpose	Employment, Reason for Travel
Intention	Plans	Events, Locations
Intention	People	Relations, Friends, Colleagues, Mentors

Table 3.1: CCE topics and associated aspects of those topics covering identity and intention.

The first step in the iterative section of the CCE process is topic selection. CCE prescribes that topics should cover a mix of identity and intention. In a job interviewing scenario, topics will most likely be selected from a CV, whereas in security interviewing topics will be predefined. In both cases, topic order should be randomised to enhance unpredictability, although job interviewing is somewhat constrained by interview norms, such as starting on the topic of your current employment. Topic examples are shown in Table 3.1, along with the aspects of each topic that might be discused in a topic iteration.

3.1.3 *Information Gathering Questions*

Information gathering questions (IGQs) are used in CCE partly for gathering knowledge for the purpose of controlling the interview, and partly for committing the interviewee to a version of the truth which can then be tested. IGQs are based on aspects of a topic, such as the examples in Table 3.1, and should ideally cover the past, present and future plans. IGQs should be in the TED format (Tell, Explain, Describe) which encourages open questions or at least questions with multiple potential outcomes. The IGQ is an opportunity for the interviewer to keep the interviewee talking while listening for ideal information on which to base a test question. The CCE principal of unexpected questioning should be followed here, to steer the interviewee away from any practiced dialogue. This steering is essential to prevent the interviewee talking at length about tangential areas they are comfortable with, but which do not provide testing oppor-

tunities. The interviewer must control the interview to ensure their desired questions are answered.

3.1.4 *Test Questions*

Test questions (TQs) are used by CCE to make use of information gained from IGQs, aiming to test the account while observing interviewee behaviour. TQs should be in the 5WH format (Who, What, Where, When, Why, How) and, as with IGQs, should be open or focussed, avoiding closed questions with minimal information return. TQs should be verifiable *in theory* as interviewers cannot be expected to know anything about an interviewee's hobby, skills, knowledge or interests. Additionally, an interviewee does not know how knowledgeable the interviewer is on a subject, so deceivers must fabricate as they see fit, causing increased cognitive load. CCE as standard does not rely on veracity checking against answers, but focusses on the observation of behaviour change against a baseline to detect deception.

Test questions should be unexpected tests of expected episodic knowledge where possible, although more detailed semantic knowledge is frequently useful. For example, if discussing a degree course, asking "which were your favourite professors?" is arguably unexpected and testing an episodic account of a student's preferences. Asking "which was the main textbook?" is arguably part of the taxonomy of the subject and thereby semantic knowledge, but is still a detailed-enough useful question. Whereas, asking "why did you want to study this subject at university?" is predictable and it is relatively easy to fabricate an answer using semantic knowledge of the subject.

Intek's primary goal is to provide interviewer support in this area of test question generation and quick veracity checking. Episodic and semantic knowledge is compared at a high-level in Table 3.2. Unexpected tests of expected knowledge are discussed further in Section 3.6.

Knowledge	Attributes
Semantic	1. "General" knowledge, known to many people 2. Usually structured/taxonomic 3. Can be acquired through study 4. Can be lacking in rich detail
Episodic	1. Particular to each individual 2. Stored temporally 3. Acquired through personal experience 4. Rich in detail

Table 3.2: The main comparitive attributes of semantic and episodic knowledge.

3.1.5 *Evaluation*

The aim of a CCE topic-based iteration is for an interviewer to reach a decision on whether they are confident in the interviewee's account. Typically two or three tests are required per topic for the interviewer to get a sufficiently detailed and temporally cross-referenced account. For the interview overall, three to six topics might be covered, depending on the time available, before an overall judgement is made.

Table 3.3 gives some example dialogue from the different phases of a CCE interview.

CCE Phase	Dialog
Baseline	• Er: Hello, how are you? • Ee: ... • Er: How is the weather out there? • Ee: ... • Er: Did you have any problems setting up the meeting? • Ee: ...
Iteration 1 Test 1	• Er IGQ: Tell me about your work. • Ee: I am a student, studying Chemistry at Lancaster University. My dad, who runs a road transport company in China has paid for my course. • Er TQ: What branch of Chemistry are you going to specialise in? • Ee: ...
Iteration 1 Test 2	• Er IGQ: Tell me about your career development plans. • Ee: I'm planning to get into mobile application design for companies. • Er TQ: What qualifications do you need to do that then? • Ee: ...
Iteration 2 Test 1	• Er IGQ: Describe your hobby to me. • Ee: I am a guitarist, played with quite a few bands, Bruce Springsteen was probably the career highlight. Played in his North East concerts in 85, when his regular East Street Band guitarist got sick. • Er TQ: Interviewer: What song did Bruce always start his encores with? • Ee: ...

Table 3.3: Example CCE discourse showing pertinent interviewer (Er) and interviewee (Ee) dialogue (not exhaustive).

3.2 INTEK OVERVIEW AND RELATION TO CCE

This section describes how the key elements of CCE relate to Intek and introduces some new terms for Intek: information snippets and factoids.

Intek is designed to support the CCE process by iteratively extracting and displaying IGQ and TQ information, based on a search performed by an interviewer by specifying a topic type and an associated snippet of information. These information snippets can be extracted from documentation accompanying the interview, such as a CV, or elicited from an interviewee during an interview by the interviewer. TQ information is displayed in the form of factoids, which display facts in various formats. This TQ information may be used by the interviewer during the interview to generate TQs.

We now describe Intek key concepts.

Topics are generally equivalent between CCE and Intek, in that they both act as a "container" for information gathering and test questioning. Intek supports six topics identified through analysis as the most frequently used, useful and feasible: Home; Organisation; Job; Tool; Course; Interest. In the Intek user interface (UI), topics physically contain IGQ and TQ information grouped around coherent aspects of a topic, such as summary, travel or points of interest. Topics also act as the fundamental building blocks of the interview in the UI, allowing the interviewer to keep track of progress and jump between topics as required. The identification of the topics supported by Intek is discussed in Section 3.6.1 and 3.6.2.

A single **Information Snippet** must be supplied for each topic in the Intek UI. The information extraction (IE) routines base their searches on this snippet and the selected topic type, with the aim of extracting facts suitable for unexpected tests of expected knowledge. The meaning of "unexpected tests of expected knowledge" is examined in detail in Section 3.6.

Information Gathering Questions are equivalent between CCE and Intek. Intek supplies static topic-specific IGQs grouped around coherent aspects of the information extracted for TQs. In Intek, IGQs are designed to lead smoothly into the TQ information in that group for natural test questioning. IGQs can also be used on their own to gather information for an interviewer's own test questions if for some reason no TQ information is extracted or useful. Examples of IGQs are shown in Table 3.3.

Intek delivers **Factoids** containing one or more facts, displayed in one of various formats, for example tables, text paragraphs, graphs or images. The interviewer must identify which facts they wish to ask about, then mentally transform the fact into an appropriate **Test Question** before asking the question. CCE dictates that questioning should ideally switch between past, present and future at random to maximise cognitive load for deceivers that must maintain an increasing fabricated story for each topic. With Intek, we find the past is unreliable for information extraction as web sites are inconsistently timestamped, and the historic information that is available, for example events and locations appearing in historic newspaper articles, is not necessarily interviewee expected knowledge. The future is also problematic, as Intek maintains interviewee anonymity and therefore does not access any social media information potentially containing specific future plans. General information that might be extracted about future plans is quite vague and also not necessarily interviewee expected knowledge, for example potential travel locations or accomodation. Intek therefore focusses on present information only, or more correctly, assumes that all web site content is present information. In order to compensate for this lack of temporal contrast, Intek aims to build deceiver cognitive load by presenting contrasting and unexpected information modalities, such as visual descriptions, geographical points of interest and directions, procedure knowledge and key people as well as plain facts. These different aspects of a topic are included in Intek development based on an assessment of the quantity and quality of the episodic or detailed-semantic information they contain and the technical feasibility of extracting that information. Section 3.6 contains more detail on factoid selection and extraction.

Factoids are displayed in the UI, along with related IGQs in **Factoid Groups** based around coherent aspects of a topic similar to those shown in Figure 3.1. All the above elements can be seen in an Intek screenshot in Figure 3.2.

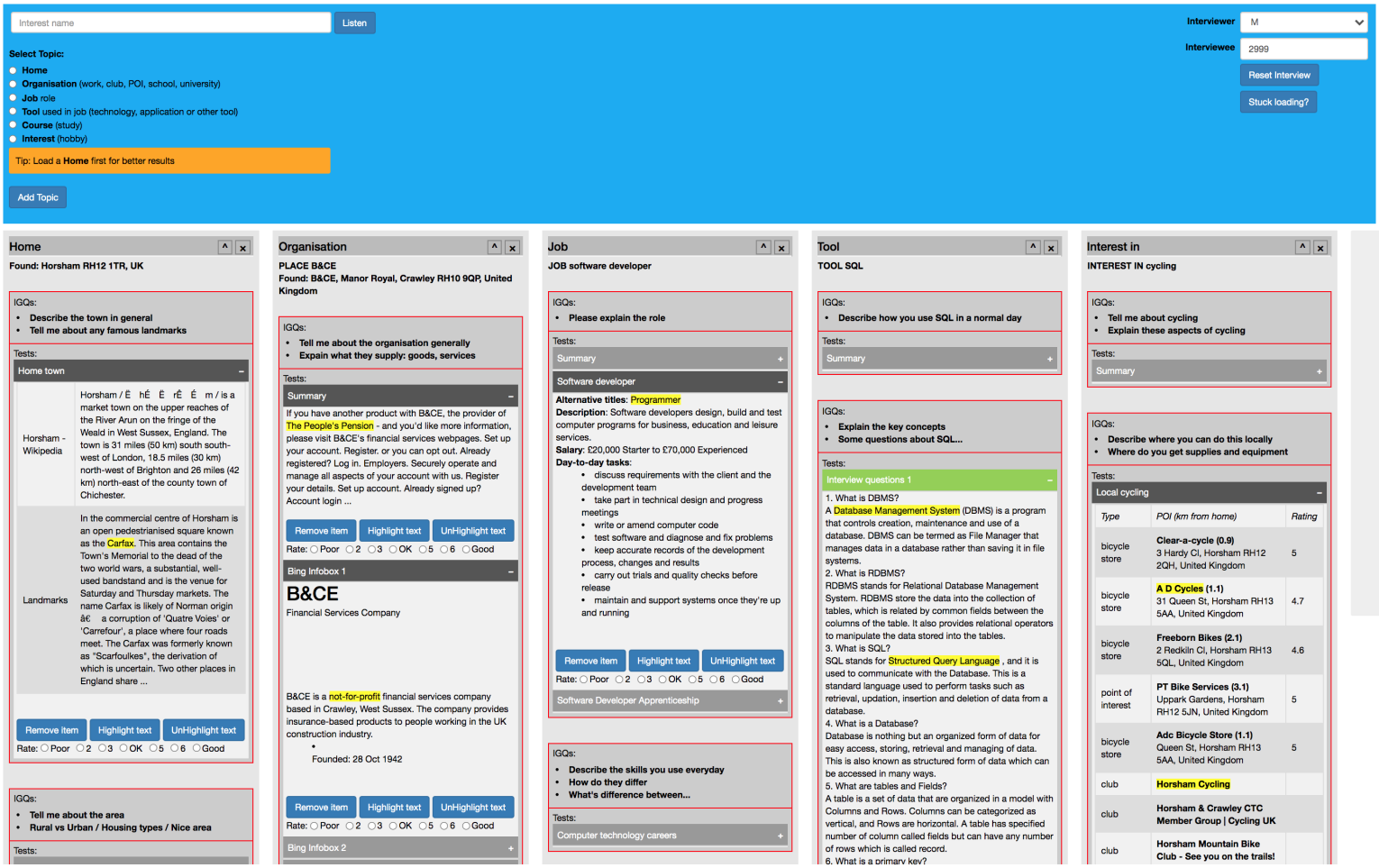

Figure 3.2: The Intek user interface with five topics loaded and stacked horizontally. Factoid groups containing IGQs and TQ information are bordered in red within each topic. Some test questions have been highlighted in advance in yellow.

While CCE and Intek are designed for deception detection in any interviewing scenario, Intek has been tailored for the task of job interviewing, as this is the focus of our main study in Chapter 5. While this tailoring does not change the overall design or Intek's applicability to interviewing in general, functions have been added to support the extra time available in job interviewing to prepare an interview beforehand from a CV, such as the ability to highlight specific questions to ask.

We now move on to discuss the development of Intek, starting with the development life cycle.

3.3 DEVELOPMENT LIFE CYCLE

This section introduces the development life cycle used for all aspects of Intek development, both the user interface and information extraction elements. Both the user interface development Section 3.5 and factoid information extraction Section 3.6 are structured using the stages in this life cycle.

Intek is a system based on a potentially impossible idea, that of extracting personal questions or questions that approximately test an interviewee's episodic knowledge, but from in-personal public data sources. Quick iteration of requirement gathering, idea generation, prototypying and evaluation was essential to explore whether Intek was initially feasible and eventually implementable within a reasonable timescale. We used a simple iterative development life cycle (see Figure 3.3) with multiple levels of user involvement and evaluation. Many iterations were undertaken in the development of Intek, but in general, the initial stages focussed on establishing higher-level requirements and design, the intermediate stages focussed on prototyping and evaluation of new topics, fact extractors and UI elements, with the final stages focussing on user testing, productionising and the final study.

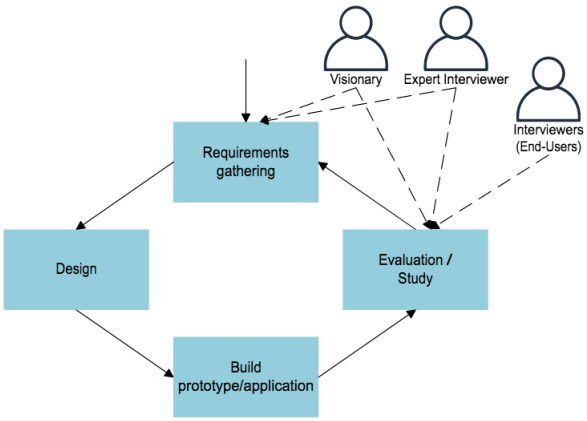

Figure 3.3: A high-level view of the iterative Intek Development Life Cycle and user types involved.

The techniques and resources used to inform each stage in the life cycle are as follows. These are explored in more detail in the UI Section 3.5 and IE Section 3.6.

1. Requirements gathering

a) High-level and detailed requirement gathering interviews were performed with the "Visionary" and "Expert Interviewer" individually and together. This provided a range of insight and views on which to base initial high-level design and to discuss functionality when it was released for prototyping.

b) An entity relationship analysis was undertaken to capture static elements of the interviewing scenario for high-level design.

c) A hierarchical task analysis was undertaken to capture dynamic elements of the interviewing scenario to inform UI design.

d) An analysis of transcripts from a previous CCE study was performed to establish a foundation of likely good topics and associated facts.

2. Design

a) Overall design rules were extracted from our high-level requirements.

b) Suitable usability rules were identified to inform UI design.

c) Rules were established from our high-level requirements for the identification and evaluation of IE sources.

d) High-level design decisions were made which influenced the direction we took.

3. Build prototype/application

a) Prototyping was used to demonstrate new UI functionality.

b) Prototyping was used to evaluate the feasibility of new factoid data sources, their extraction and integration into the UI.

4. Evaluation/Study

a) Prototype evaluation interviews were undertaken with the Visionary and Expert Interviewer which informed the further development or rejection of functionality.

b) End user testing was performed to identify any potential issues hindering our novice interviewers.

c) Public demonstrations of Intek generated high-level feedback.

d) The formal interviewing study in Chapter 5 gives bottom-line results for deception detection task performance.

The effort involved in creating Intek can be split into two main areas: developing a simple intuitive UI for interviewer support and also identifying and extracting useful factoids that interviewers can use to form test questions.

The Intek UI consists of a single page that expands significantly as new topics are searched, intuitively guiding the user into following the ideal CCE process of iterative topic-based discussion and veracity-checkable test question delivery.

The extraction of useful factoids is the most time-consuming and complex area of development. We formulate and apply rules for the identification of topic-based fact sources, then develop IE routines to extract the most pertitent information from these sources. This task can be straightforward in the case of extracting from an API, for example Google Maps information for nearby points of interest, or complex in the case of our named entity recognition (NER) and extraction pipeline, which attempts to extract people, locations, events, costs and other information that an interviewee should know about an organisation they have close contact with. This NER pipeline employs multiple levels of IE and NER, marshalling state-of-the-art NER routines and is discussed further in Section 3.6.4.3.

3.4 HIGH-LEVEL DESIGN

This section details critical design decisions that were taken early in development as a result of initial requirements and early prototyping.

3.4.1 *Front-End and Back-End Technologies*

The technologies used in Intek development are not surprising. For the back-end we use Python, as this is widely supported by many developers, for many conceivable scenarios, including state-of-the-art NLP libraries. Python is also a speedy flexible language for prototyping as well as being ideally suited for dealing with the many granularities of data required for an application implementing NLP and IE technologies. The use of Python supports our requirements around library re-use, quick prototyping and evaluation. For the front-end we use web/HTML as this is clearly a robust and flexible technology

supporting our requirements around remote access, UI flexibility, interactivity and accessibility.

3.4.2 *Pro-Active vs Re-Active Deception Detection*

This choice refers to pro-actively supplying information for improved test questioning as opposed to re-actively detecting the signs of deception. Intek is designed around the pro-active appoach.

In theory a re-active approach would rely on the interviewer generating test questions, as in a standard CCE approach, but offering veracity and behavioural-based judgements to the interviewer after test questions were answered by the interviewee, either as soon as possible or at the end of the interview. The re-active approach alone has a number of drawbacks. It does not support requirements around aiding the interviewer in test question generation, which is seen as one of the key reasons for poor deception detection performance in literature (Ormerod and Dando, 2015). A re-active approach does not explicitly need to support the interviewer in the centralisation of interview research or keeping track of the interview, which are key parts of the assistance Intek affords interviewers. As previously mentioned, behavioural-based deception detection has been shown to perform poorly, although linguistic techniques show more promise. Veracity-based claim checking, although performing relatively well in a general/public scenario, may be very difficult to accomplish for finer-grained facts that probe episodic knowledge. Also the detection and segmentation of claims to be checked as well as source identification for checking would be challenging tasks.

For Intek, we implement a pro-active approach to interviewer assistance by supplying a good range of test questions in response to topic-based snippets of information from the interviewee. This should immediately remove some of the cognitive load from the interviewer, by relieving them of the task of thinking of novel questions; interviewers simply select the information they want to discuss. This approach has the additional benefit of encouraging interviewers to follow the approved CCE process: asking a range of tests across different aspects of the topic using different information modalities and keeping track of the interview structure and process overall. The delivery of good test questions improves the likelihood that deceptive behaviour will occur in deceiver interviewees and thereby deception detection. The pro-active approach supports three of the six high-level

requirements that re-active does not: supporting the CCE process, informed conversation rather than interrogation and decreasing interviewer cognitive load.

For further development of Intek in the future, it might prove fruitful to incorporate some elements of the re-active approach into the existing system, for example linguistic analysis might be added to indicate changes in interviewee dialogue against baseline.

3.4.3 *Questions vs Factoids*

This section refers to the delivery of fact-based test question information in the UI. Should Intek deliver a list of formatted questions ready for asking, or present information formatted for readability according to the type of information being extracted. For example, a summary paragraph might be formatted into a list of questions in the former approach, or displayed as a paragraph in the latter approach. The latter "factoid" approach should improve the efficiency of interviewers *understanding* the paragraph.

An approach using lists of questions suffers from a number of drawbacks. Question lists were shown in initial testing to be more taxing and slower to read and understand contextually, particularly where multiple facts exist from a single source. The interviewer becomes a "question robot", simply asking questions without the time to understand the subject behind the questions. Technically, another layer of processing would be required to extract facts, then format them as questions and answers. However, the list of questions approach may be applicable in the future, to an online multi-choice interviewing scenario without an interviewer.

The factoid approach, which Intek uses, displays each set of facts for readability according to the type of fact. Testing showed humans can quickly and easily deliver facts as a question. Factoids give interviewers a better sense of context around each set of facts, supporting the requirement for knowledge-based conversational interviewing rather than interrogation.

3.4.4 *Dynamic Display by Relevance vs Static Display by Natural Interview Order*

This choice refers to the layout and order of factoids after a search is performed. It is conceivable that the whole of Intek could be run

off a single dynamic list of factoids (or questions, had we chosen that route). In response to a topic search IE would be performed as normal, the results would then be assessed for relevance and the most relavant added to the top of the list. The interviewer then simply asks the top factoid. In contrast, the static approach uses a predictable pre-defined layout of factoid groups for each topic, based on a natural interview order: summary, high-level demographics, then more detailed tabular facts and Q&As.

The dynamic approach is reliant on an accurate relevance measure, as irrelevant questioning would clearly not aid in either detecting deception, the interviewers control over the interview or interviewer credibility. As the dynamic approach would update the order of factoids in the list with every search, this gives little opportunity for an interviewer to screen factoids for a good fit into the current interview dialog, they would have no idea of the context. This approach is clearly in line with the "clutter-free" design principle, but takes this to extremes, as the lack of a consistent position or grouping for factoid subject leads to an interviewer less aware of topic and interview context, violating the requiment for the interviewer to lead a knowledge-based conversational interview. In early prototypes, the dynamic method was found to be cognitively demanding for the user, due to the constantly changing nature of the UI and the frequent mental updates required.

The static approach, which Intek uses, orders topic-based factoid subject groups by a more natural interview sequence, starting initially with high-level summary-based information and demographics, followed by more detailed factual questioning and lastly more speculative Q&As. This sequence always remains the same for each topic in line with the consistency design principle. These predicatable structures give the interviewer a better sense of context within the interview as a whole and within topic subject. A potential downside is the sheer amount of information on screen at times; topics can grow vertically to take up as much space as required. This is mitigated by collapsible factoids which only need to be kept open if being used or planned for questioning. We also implement a heuristic relevance measure which highlights potentially useful factoids as a guide. Early prototyping with large vertical topics stacked horizontally, as in Figure 3.2, indicated users found this layout natural and intuitive, much like reading any vertically-scrolling web page.

With these high-level decisions made, we now move on to development of the user interface and then information extraction processes.

3.5 INTEK USER INTERFACE DEVELOPMENT

This section details the main elements of the Intek UI and shows how they evolved as a result of evaluation and iteration. The sub-sections reflect stages in the development life cycle: requirements gathering; notational analysis; user interface design; implementation and evaluation.

3.5.1 *Requirement Gathering*

Interviews were used for initial "brainstorming" and high-level requirement gathering, ongoing refinement of requirements, and for the evaluation of prototypes and the final Intek product. All stakeholders were interviewed at some point, either individually, in group sessions or both, to harness the varying dynamics of those situtations. The level of feedback elicited differed with the type of stakeholder, generally the "Visionary" at a higher-level, end-users at a lower-level. Intially more open discussion was encouraged, which become more focussed in later iterations as the design became concrete.

The Visionary is an expert on the CCE process Intek supports and also has wide experience in the psychology and practice of interviewing. The Visionary was consulted for high and medium-level requirements, design input around the Intek process and to some extent how the process was implemented in the UI. The Expert Interviewer has wide experience of standard (non-CCE) job interviewing and implementation of CCE in a job interviewing scenario. The Expert Interviewer was consulted for medium and low-level issues around the practicality of presentation and information returned. The Visionary and Expert Interviewer were frequently consulted as a group (the "Expert Group"), especially leading up to and during the final study.

The final refined requirement list is as follows. Note the initial high-level requirements, which break down into more concrete functional and non-function requirements and constraints. The MoSCoW notation follows each detailed requirement indicating priority.

1. Increase deceiver-interviewee cognitive load, while decreasing truth-teller-interviewee cognitive load. This is the essential aim of the CCE method which Intek supports.

a) Support the interviewer in following an iterative topic-based process. (M)

b) Support the interviewer in delivering information gathering questions. (S)

c) Support the interviewer in delivering test questions for each topic. (M)

d) Support multiple topic types relevant to the interview scenario, ideally covering identity and intention. (M)

2. Information must be delivered in real-time. The technology should not impact the natural flow of the interview.

   a) Real-time delivery, the interviewer must be able to start asking questions within a few seconds of a search. (M)

   b) Do not overburden the interviewer with input tasks. Ideally one simple input event per topic. (M)

   c) The technology should not be a factor in the interview from the interviewee's perspective; the technology is covert. (M)

3. Support interviewers in generating good CCE test questions and delivering a good CCE interview, aim to decrease interviewer cognitive load.

   a) Information delivered should be fact-based, thereby providing an answer as well as a question. In this way questions are veracity testable in real-time. (M)

   b) Deliver a good variety of information that can be used by the interviewer to generate unexpected tests of expected knowledge. (M)

   c) Quality not quantity; information should be relevant to tests of episodic knowledge, not too general, not too obscure, that can be parsed in real-time by the interviewer. (M)

   d) Indicate the most pertinent/relevant information for a topic to help the interviewer identify the best information as quickly as possible. (S)

   e) Allow the interviewer to highlight information in preparation that they wish to ask in the interview. (S)

   f) Allow quick and easy question selection and delivery. (M)

g) Deliver information in a format that is easily parsed in real-time by the interviewer. (M)

h) Allow the interviewer to easily keep track of overall interview progress. (M)

i) Allow quick and easy back-tracking to previously un-asked questions for further questioning if required. (S)

j) Consistently format the presentation of topics and information for quick parsing. (S)

k) Create a simple straightforward UI supporting the CCE process. (M)

l) Centralise all information gathered in one place. (S)

4. Deliver some familiarisation/summary information with each topic. Give interviewers the chance to quickly understand each topic and thereby engage in a fact-based conversation, rather than reading a list of prepared questions verbatim.

   a) Deliver specific factoids that contain contextual/background information for each topic. (M)

   b) Provide a range of topic-based information, depth and breadth. (S)

5. Design to support a broad range of investigative interviewing scenarios, whether prior interviewee information is available or not.

   a) Support interview preparation in advance. (M)

   b) Support real-time questioning with no preparation. (M)

   c) Speech-driven input option for hands-free environments. (C)

6. Interviewee anonymity and flexible accessible development methodology.

   a) No information will be supplied to the system that might uniquely identify an interviewee. (M)

   b) Do not re-invent the wheel, use suitable off-the-shelf libraries if they exist. (S)

   c) Demonstrate, iterate, negotiate, fail fast. (M)

   d) Full logging to support future analysis. (M)

e) System should be accessible remotely by mutiple concurrent users. (M)

These requirements (both functional and non-functional) input into the design and implementation of the UI, IE and other back-end system functions. As we discuss the development of Intek we will provide links back to the relevant supporting requirements. As this is a novel system, experimental in nature and not safety critical, all requirements are somewhat flexible based around the feasibility of implementation.

3.5.2 *Notational Analysis*

In order to more formally explore Intek functional requirements, we employed two methods of notational analysis: an entity relationship diagram (ERD) to capture static structural entities and their behavioural relationships; a hierarchical task analysis (HTA) based on user-goals to capture dynamic behaviour and system-user interactions. This formal analysis is particularly relevant for Intek as it is a novel application with no existing system to work from. These analyses are also a foundation to align the designer with the domain experts with regard to terminology and domain understanding.

The ERD shown in Figure 3.4 provides a good basis for UI design, especially around the structure of topics. Each topic requires key information from the interviewee, which is then searched by the interviewer. Topics should be structured into groups of factoids containing similar information along with introductory information gathering questions. The ERD also forms a good basis for a data model, indicating which information needs to be persisted for which entities and granularities, for both application operation and usability analysis logging.

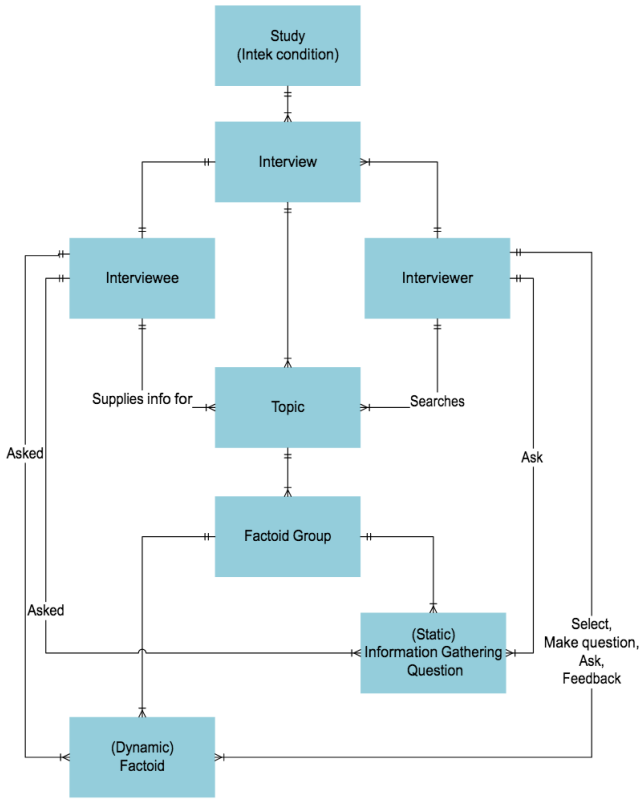

Figure 3.4: Intek entities and their relationships

Hierarchical task analysis with sub-goal template (Richardson et al., 1998; Ormerod, 2000; Ormerod and Shepherd, 2003) is a good method for fully capturing detailed dynamic system interactions for UI design, especially for new systems, by focussing on the sequence of user actions toward user goals in full detail. The Intek HTA was initially created by the Visionary to handle standalone CCE interviewing, then amended through discussion to include Intek system interactions.

The Intek HTA (see Appendix 1 Figure 7.1) contains six high-level stages that must be performed by an interviewer: interview planning; conduct a baseline; ask information gathering questions on a selected topic; ask test questions; end the interview; evaluate the outcome. Each stage was examined in detail to ascertain the system interactions required to achieve each goal. We are left with a small list of unique interactions which align with our functional requirements to provide the essential UI elements in sequence. These are the critical interviewer interactions Intek must support.

1. Select an interviewer and interviewee to identify the interview.
2. Enter the desired information snippet.

3. Select a topic and search.
4. Select an IGQ for delivery (from a pre-defined topic-specific list).
5. Select a TQ for delivery (from the information delivered by Intek search).
6. Feedback on TQ quality.

3.5.3 *User Interface Design*

Intek user interface implementation is based on guidelines sourced from the requirements above and an updated version of Nielsen and Molich's user interface design guidelines (Molich and Nielsen, 1990; Wong, 2020) which cover the usability, utility and desirability of UI designs.

The following design guidelines are based on our requirements. Each guideline is listed with the supporting requirements from Section 3.5.1 in parentheses.

1. Visually support the CCE process: iteration; hierarchy of topics, IGQs and factoids. (1a, 3k)
2. Quick feedback from user actions and visibility of long running processes. (2a)
3. Simple user-system interactions. Minimise mouse clicks. Allow keyboard shortcuts if possible. (2b)
4. Support hands-free operation (with speech input). (5b, 5c)
5. Support the natural interview flow from conversation into to testing. Group easily comparible alternative facts if possible. (4, 4b)
6. Present potentially large amounts of information in a simple minimalist design. Do not over burden working memory. (3, 3b, 3c, 3f, 3k)
7. Some unwanted information is likely to be returned from search. Include options to highlight/indicate useful infomation, while allowing unuseful information to be hidden or entirely removed. (3c, 3d)

8. Consistent formatting and metaphors internally (between different topics) and externally (elements to use standard web paradigms). (3j)

9. Use as few pages for display as possible. If multiple pages are used, all other pages should be easily referable from the initial page. (3l)

The following section lists Nielsen and Molich's ten user interface design guidelines, which were used along with the above guidelines from our own requirements to guide Intek UI design. In the following paragraphs, we include details of the degree to which Intek eventually implemented them. All the UI functionality mentioned is visible in the Intek UI screenshot in Figure 3.2.

Visibility of system status: provide feedback from user actions and give an indication of how longer running processes are progressing. Intek applies this by disabling submit buttons once pressed, changing wordings to indicate what action has been taken and using status colours for longer running asynchronous processes. Intek could have gone further by implementing progress bars for these longer running processes, but this would have taken significant time to implement and may have distracted interviewers while they dealt with the task of interviewing.

Match between system and the real world: use terminology that users understand, both in the CCE domain and in general. Intek consistently uses terms that match with the CCE literature and the user training session. In general, Intek terminology is non-technical, such as the button labelled "Stuck loading?" error recovery function.

User control and freedom: allow fine control of actions where necessary and include the ability to reverse actions taken. Intek actions are all visible and repeatable. Actions that result in processes firing can be stopped or closed. Intek does not include an explicit "undo" function, but this would add little functionality given the existing repeatability and removability.

Consistency and standards: consistency of design, element types and terminology across different application pages and versions. Intek contains only a single page and single version, using standard web elements and page design.

Error prevention: reduce the chance of user mistakes by explaining system options so users understand their actions. Intek includes helpful tooltips, tips and labels. We use collapsible elements with helpful

titles to contain the bulk of on screen information, this allows the user to understand the contents before opening the element, as well as reducing UI clutter. User testing caught and fixed a handful of issues that caused user errors. For example, the initial order of topic selector then information snippet textbox in the UI caused users to enter the snippet but consistently forget to select a topic; reversing the order of these two elements solved the problem.

Recognition rather than recall: do not rely on user memory when the application can present a list of options. Intek explicitly lists options rather than relying on knowledge of CCE. Intek initially used a single textbox for topic search, relying on users to type the topic and the information snippet in a single line of text. This approach caused much confusion, despite the explanatory information provided, leading to the current radio topic selector and separate textbox for information snippet.

Flexibility and efficiency of use: implement shortcuts and recording of efficient sequences of actions to improve efficiency for experienced users. Intek allows the keyboard to quickly be used to select topic, enter text and search in sequence. Screen scrolling can be done with mouse or keyboard. The speech interface can be used to minimise interaction down to a few key presses, even for longer text snippets. We chose to display all factoid groups in the UI even if the factoids in that group are empty (where no information could be extracted); this could be seen as an inefficient use of screen real-estate, but after expert evaluation the benefits of a consistent layout and the inclusion of potentially useful IGQs (IGQs are static and always displayed) outweighed the potentially negative impact of the clutter.

Aesthetic and minimalist design: try to reduce user cognitive load with a minimalist design, reducing screen clutter. For Intek this is especially important for topics which contain a lot of information. All factoids are contained within collapsible elements, thereby only opened elements significantly clutter the UI. Consistent factoid group containers (red borders) are used, which aim to maintain group cohesion whether collapsibles are open or closed. By nature of the few actions available to the user (see HTA above in Section 3.5.2), the UI is minimalist. We found the simplest possible left-to-right topic stacking method preferable for allowing users easy access to all potential test questions, re-cap previous topics, follow the interview flow and press interviewees in areas which they deemed suspicious. We added functionality for minimising stacked topics.

Help users recognize, diagnose and recover from errors: if something goes wrong, tell the user what went wrong and how to fix it in an understandable way. Intek uses friendly error messages (that are also logged) for frequent errors that are unfixable by the user, such as timeouts. When an unknown error occurs, Intek tries to auto-recover to a position where the user can continue as before. If asynchronous processes fail to return, the "Stuck loading?" function can be used to continue. Intek does not explicitly tell users how to recover, as in theory they will always be back in the same position they were before the error.

Help and documentation: if help is required, it should be easily located. Intek offers tips and clearly labelled actions. An initial hands-on training session was provided for interviewers. Help is supplied in a separate document used in these user training sessions.

3.5.4 *Implementation and Evaluation*

We now discuss the main features of the Intek UI and show how these have evolved over the project life cycle as a result of feedback. These features are: topic selection and the supply of an information snippet; topic stacking; factoid groupings; factoid types; factoid status and relevance; real-world use in our interviewing study.

3.5.4.1 *Topic selection and supply a snippet of information*

The simple action of selecting a topic and supplying a small piece of information on which to base a search is one of the key user actions in Intek. Initially, the design was to rely primarily on speech input, so the topic and information snippet were supplied in a single text box as in Figure 3.5. However, it became clear after testing that even with format guides in place, users were not sure what to type in the box. Therefore, the design evolved to separate out the topic selection to a list of the supported topics, initially in a drop-down format, as in Figure 3.6, but later to a radio list with all options visible without an unnecesary extra click, as in Figure 3.7. Finally, the orders of topic radio selector and text box were reversed after end-user testing revealed users repeatedly ignored topic selection if it was placed first, see Figure 3.8.

Figure 3.8 shows, to the right of the screen, selectors used to indicate the interviewer and interviewee, these are both anonymous identifiers used primarily for logging for later analysis. The "Reset

interview" function is used at the end of an interview to clear the screen ready for the next interview; browser refresh can also be used. The "Stuck loading?" function is used to recover from timeouts of long-running processes, this is a rare occurance caused by many concurrent users especially in the initial training sessions.

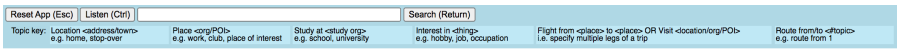

Figure 3.5 The initial single box approach, which includes both topic and snippet. The light blue boxes contain format guides.

Figure 3.6 Topic selection is moved to a drop-down selector, with a separate text box for the snippet.

Figure 3.7 Topic selection is changed to a more visible radio selector.

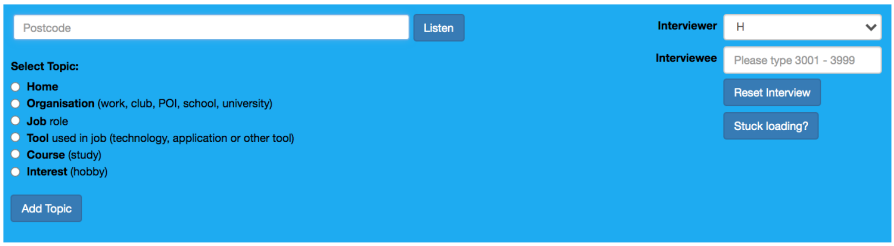

Figure 3.8 In the final version, topic selection and text box order is reversed as a result of feedback.

3.5.4.2 *Topic stacking*

The layout and accessibility of topics is an essential element of Intek's usability for interviewing. Topics need to be individually recognisable for cohesive conversation and test questioning, also a topic's position within the interview as a whole should be easily assessible. Intek uses a simple horizontal topic stacking method, where new topics are appended to the right of existing topics (see Figure 3.9). In early designs, topics were stacked vertically as well as horizontally up to a height limit, but testing showed this approach to be much more difficult to operate, partcularly due to the extra dimension to navigate around during an interview and also due to the dynamic

resisizing of topics as factoid collapsibles are opened and closed resulting in unpredictable vertical topic positions. Figure 3.10 shows the Tool topic "minimised" in-place; this functionality was added to reduce horizontal space requirements and clutter once a topic had been used for questioning. Testing showed users rarely used this function indicating the simple stacking system works well. A more complex "toolbar" minimisation system was considered, but user feedback indicated this would add complexity for no great benefit.

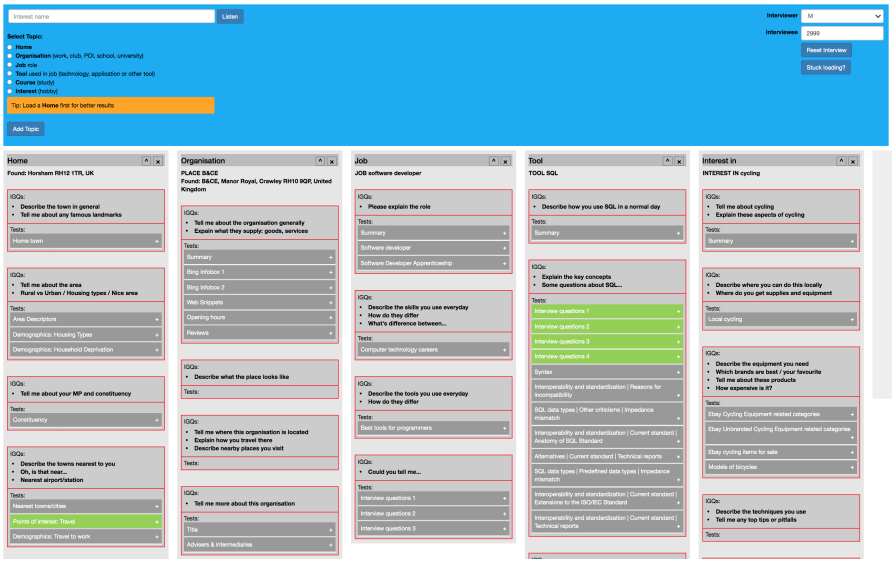

Figure 3.9 The Intek user interface with five topics stacked horizontally.

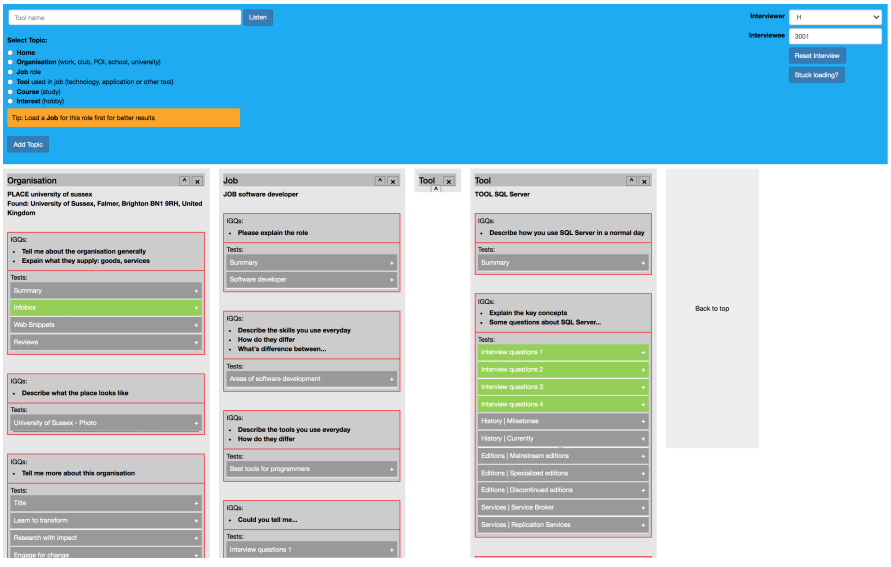

Figure 3.10 The Tool topic has been minimised to reduce horizontal space requirements.

3.5.4.3 *Factoid groups*

Within each topic, factoids containing test question information are grouped around similar aspects of that topic, for example travel information as in Figure 3.11. Each factoid group evolved to contain a distinct and coherent combination of fact sources that could sensibly be grouped with IGQs to provide varying and unexpected aspects of interviewee expected knowledge. Factoid group order within a topic losely follows a natural interview sequence, starting with conversational summary and high-level demographics information, then moving into fact-based tabular and list-based factoids usually with multiple alternative facts, then finally into text-based single Q&As. Each factoid group has a small number of associated IGQs, which aim to stimulate high-level conversation in an area that can lead naturally into the associated test questioning. IGQs are pre-defined and remain the same for each factoid group. These IGQs have been generated through discussion and scenario-based testing with the expert interviewer and refined through end-user testing.

The link between IGQs and factoid groups has evolved from the early "cue questions" (see Figure 3.13), which were located at the end of the topic and could be dynamically added to by the user. End of topic was poor positioning for their intended use for leading into related test questions and the "Add Question" function became irrelevant after a good set of IGQs had been established. The next evolution imposed a strict sequence of IGQ selection by the user which then highlighted the relevant associated factoids (see Figure 3.14). This system is appealing in theory as it forces adherance to CCE process, but in practice was too complex to operate, involving several extra interactions, causing users to ignore the IGQs and move straight to the factoid test questions. The final design (see Figure 3.9) separated factoid groups and their constituent IGQs and factoids with double red borders. This approach made it very simple for users to deliver the relevant IGQs for a factoid group. Importantly, the double border maintained the cohesion of the factoid groups as they dynamically resized as a result of opening and closing factoid collapsibles (compare Figures 3.11 and 3.12).

The granularity of individual factoids has evolved since the early fine-grained factoids based on a single source of information (see Figure 3.13), to a user-centred approach around lines of questioning, in which similar facts and information are grouped together to reduce uncessary user actions and UI clutter. We discuss the idenfication of

fact sources and fact extraction in the information extraction Section 3.6.

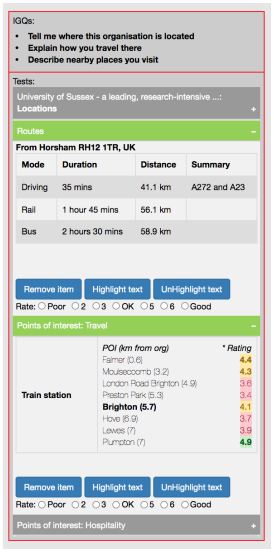

Figure 3.11 A travel-related factoid group, containing IGQs in the top section and four factoids in the bottom section.

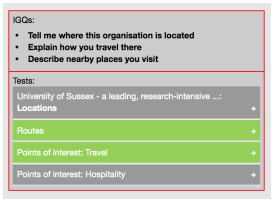

Figure 3.12 The same factoid group, with factoids closed/collapsed.

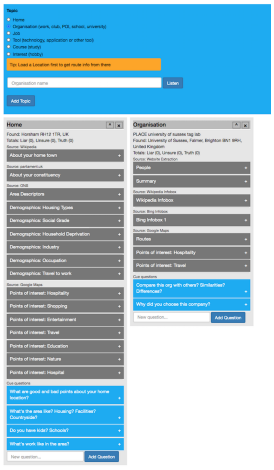

Figure 3.13 An early version of Intek with IGQs in the form of "Cue Questions" with the "Add Question" function. Factoids were fine-grained and repetitive at this time.

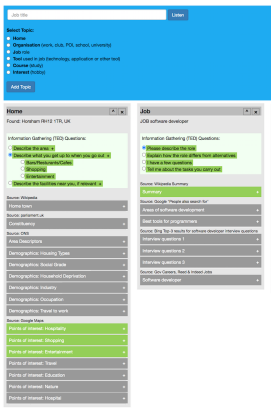

Figure 3.14 An evaluation version of Intek with IGQs explicitly linked to specific factoids, that was deemed over-complex. The green highlight indicates: 1) that factoids exist for an IGQ and 2) when an IGQ is selected, which factoids are linked.

3.5.4.4 *Factoid types*

We now detail the different types of information contained within Intek factoids. Factoids are a key part of Intek, as this information is used by interviewers to form unexpected test of expected knowledge. Factoid types are discussed in the order in which they appear in an Intek topic, although some randomisation of delivery by the interviewer is encouraged to reduce the learnability of interview sequence by the interviewee.

Most topics begin with a **Summary** extracted from Wikipedia or from an information-snippet-specific web site found via web search. Summaries are usually formatted in single-row plain text, but are sometimes tabular when additional subjects are found (see Figure 3.15). The summary is designed to quickly orient the interviewer to the topic, provide a conversational yet fact-based general topic overview and also provide some tests of expected knowledge, for example nearby places or the names of prominent landmarks in Figure 3.15.

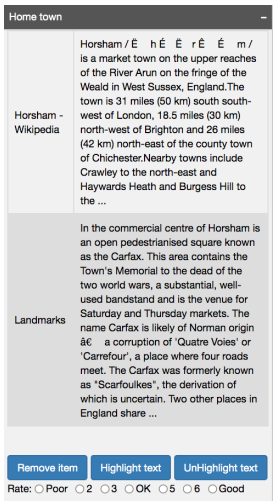

Figure 3.15 A Wikipedia summary from the Home topic, showing a general summary and landmarks.

Demographic information displayed in the form of a distribution graph, gives another high-level conversation point specific to a geographic area (postcode) or institution. The information is best used after an IGQ, for example "Tell me about the area where you live" in tandem with Figure 3.16. The interviewee is not expected to answer specific questions on percentage distributions, but should be able to generally describe their area in this way. Unbalanced graphs, such as Figure 3.16, are potentially more useful as the interviewer might be expected to mention this large disparity.

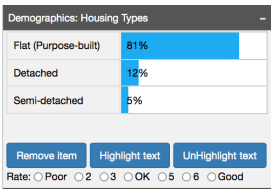

Figure 3.16 A Housing Type distribution for a postcode extracted from the ONS.

Images are displayed for Organisations where available in Google Maps API (see Figure 3.17). Images are rich in detail and usually offer good opportunities for unexpected questioning, such as "Describe the campus for me" or a more detailed "What is the name of building with two towers in the middle of campus?".

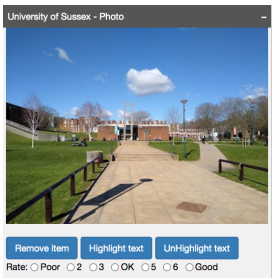

Figure 3.17 An image of the University of Sussex extraced from Google Maps API.

The **Simple List** factoid is used to display summary information, usually from Organisation web sites (see Figure 3.18). This is especially useful for smaller organisations, for example clubs or societies, where Wikipedia or Google Maps information is not available and summary details about the organisation can form useful test questions.

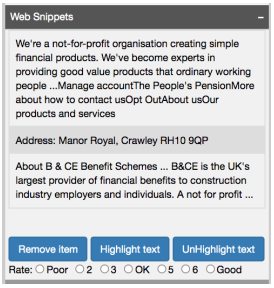

Figure 3.18 A simple list of summary information from a company web site.

The Wikipedia **Infobox** is a useful source of multiple facts about larger organisations (see Figure 3.19). The tablular infobox usually

contains a range of detailed semantic facts as well as including different and unexpected modalities, such as a logo image, which can make a great unexpected test of expected knowledge.

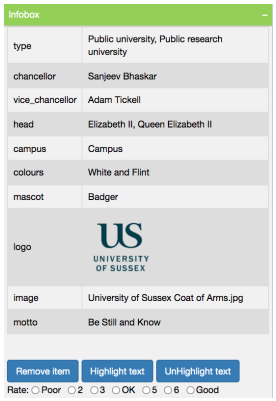

Figure 3.19 A Wikipedia "infobox" for the University of Sussex.

Geographical **Points of Interest (POIs)** have perhaps the most complex factoid type UI (see Figure 3.20). Various types of POIs are extracted from Google Maps around an interviewee Home or Organisation topic location. POIs should be well known to an interviewee, either by close geographical proximity to a very well known location or as a well known POI relatively close. The POI factoid contains several dimensions of information. POIs are broken down into type and then sub-type, for example Shopping, then Convenience Store, Department Store, Shopping Mall. POIs within these sub-types are then listed in order of distance from the Home or Organisation in kilometres, shown in parentheses. The weight of the text indicates how many reviews that POI has received relative to the other POIs; in this way the relative popularity of the POI can be gauged. The "Rating" column on the right indicates the average review score for that POI, which is then highlighted green, yellow or red to indicate relative quality of POIs. These two dimensions combined give the interviewer an idea which POIs are well known for quality or lack thereof. Finally, if a POI is discussed in an interview, it can be hovered over with the cursor to reveal a tooltip containing detailed location and other information.

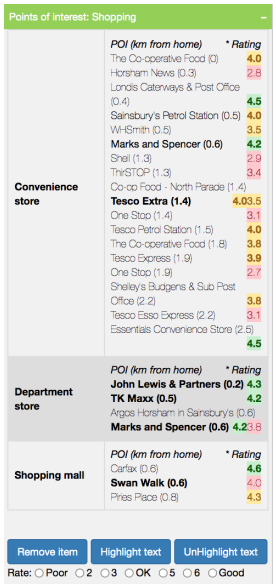

Figure 3.20 Geographical points of interest regarding shopping. Several dimensions of information are included.

The **People** factoid parses an Organisation web site for person entities and tries to extract their role and supporting text. Extracted entities are displayed in the order in which they appear in the source web site, which is typically most important first. Entities are displayed within a small snippet of contextual text which allows the role of the entity to be easily identified, in fact key roles are highlighted in green (see Figure 3.21). Each entity can be clicked upon to display the supporting text. Key people can be a good expected test of expected knowledge, particularly for medium-sized organisations where leaders will not be general knowledge, but would be known by employees and particularly for small companies were all staff should be well known. The **Location** factoid functions in exactly the same way as People, but looks for location entities.

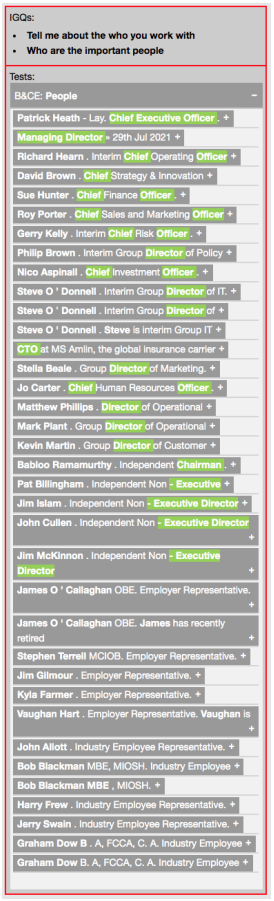

Figure 3.21 A People factoid containing a list of key staff extracted from a company web site.

Q&A factoids are extracted from Google "People Also Ask" sections and various other sources and function as frequently asked questions about a topic. Q&As are used mainly by Job, Tool and Interest topics to probe key experiential knowledge around actually performing a job, interest or using a tool. Q&As fit the collapsible UI element well (see Figure 3.22), as questions are used for titles and answers form the content; questions can be quickly scanned to find the most useful answers.

Figure 3.22 A Q&A factoid containing a question as title and textual answer as contents.

Bullet list factoids are used to display useful lists extracted directly from information-snippet-related web sites (see Figure 3.23). Bullet

list factoids are used by Job, Tool and Interest topics to extract similar information to Q&As using Bing searches such as "Interview questions" or "Common mistakes" to identify source web pages. The nature of these searches can result in lower quality web page sources, also a large proportion of these questions are not supplied with answers. Not having answers is not an issue from the CCE perspective, as behaviour change when asking the question is key, but the ability to veracity check against answers is an Intek requirement. However, where lists of alternative concepts or items within a topic can be extracted, bullet list factoids can be extremely useful for more advanced questioning techniques: elephant traps, which require a list of alternatives with which to create a negative example used as a leading question to catch out an interviewee. This technique was used multiple times in our study. Also, searches of this type can frequently deliver some information for obscure searches, where other sources fail to deliver.

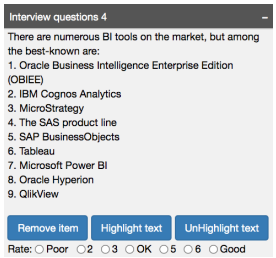

Figure 3.23 A bullet list factoid containing "interview questions" around Business Intelligence tools.

The distribution of these factoid types over the different topics used in Intek is shown in Figure 3.24.

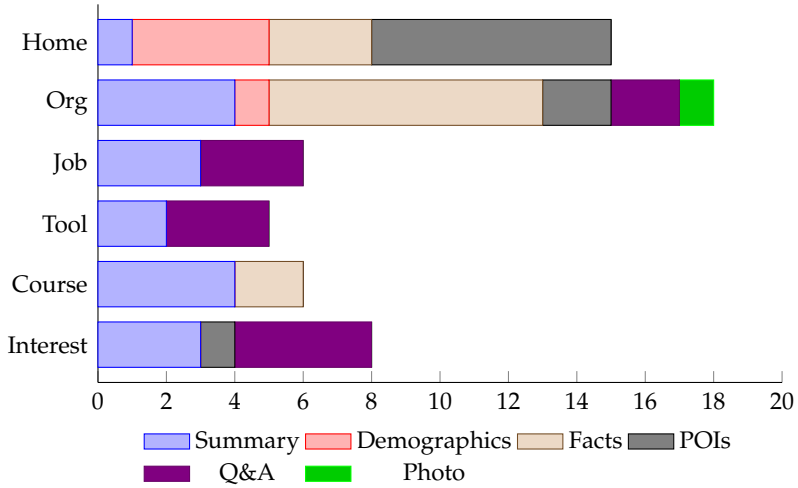

Figure 3.24 Counts of factoids available in Intek by default by factoid type and topic

3.5.4.5 *Factoid Status and Relevance*

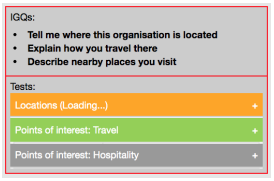

Figure 3.25 A factoid group containing a factoid still loading (orange) and a factoid deemed especially relevant (green).

Intek displays factoid status by colour in order to give a quick indication of which factoids an interviewer should open first in a real-time scenario. A standard factoid shows as light grey if closed or dark grey if opened. A factoid linked to one of the longer running pipelines, for example people, locations or summary, is shown in orange until it is fully loaded. Finally, a factoid which is deemed more relevant is highlighted green to indicate it shoud be opened first (see Figure 3.25).

Factoid relevance is currently implemented in Intek in two ways. Firstly, we manally selected factoids that have been identified through testing to consistenly contain a good range of facts which test detailed semantic knowledge and are unexpected tests of expected knowledge. Some examples of these hand-picked factoids are the Wikipedia infobox and some Interview Questions. Secondly, when multiple topics are loaded, topic cross-referencing can take place, which allows factoids more specific to interviewee experience to be extracted. Some examples of cross-referencing are Home and Organisation combining to provide route information, Home and Interest combining to provide

local outlets, clubs and location supporting that activity, Tool and Job combining to provide tool usage information specific to a role. These two approaches are naive, but provide a workable method while data is gathered for a more robust solution through the "Rate" feedback mechanism included at the bottom of every factoid. Initial investigation into the data gathered from these ratings indicates a good measure of the relevance of a factoid, for a particular topic and search, involves the relative popularity of the entity being searched versus the cumulative popularity of entities and their attributes contained in a factoid.

3.5.4.6 *Real-world usage*

Intek is designed to be used in a real-world job interviewing study, which during the COVID-19 pandemic, means supporting video conferencing (we select Zoom in our study). We recommend to interviewers in the training sessions, that preparation, to the satisfaction of the interviewer, is done in advance, thereby the only applications that must be focussed on are Zoom and Intek itself. In this scenario, Intek can be run in full-screen mode with the Zoom window re-sized to hover over the top-right of the screen. This gives the interviewer a good view of Intek and the interviewee.

We now discuss the methods by which the UI was evaluated.

3.5.5 *Evaluation*

Evaluation of the Intek UI took place in three chronological phases: main functionality prototyping with the Expert Group; end-user testing; the final job interviewing study. These phases are now examined in more detail.

Prototypes of individual design ideas incrementally added to an increasingly complete system were tested using scenario-based sessions with the Expert Group members, both as a group and individually. Two types of sessions were used: test and observation led by the developer and test alone providing feedback. Scenarios involved running through pre-determined, but realistic, topic orders usually using the tester's own CV to supply information search snippets. Feedback from these sessions was then assessed for inclusion into Intek and the life cycle re-iterated.

Intek was demonstrated to two groups, once to our laboratory colleagues and once to a public audience. Valuable feedback was re-

ceived as part of this process, particularly around the types of information potential users expected to be available.

As Intek neared completion, a range of further evaluation took place. End-users became available after the first two conditions of the study were complete, which enabled the same scenario-based testing used with the Expert Group to be undertaken with these end-users. We also employed A/B testing to evaluate alternative approaches of various UI elements, for example an easy-to-follow approach for factoid groupings. This end-user group, as novice interviewers, were less suitable for high-level input, but indispensable for feedback on the final design.

The final stage of evalution was preparation for and execution of our job interviewing study. A group hands-on training session was undertaken for all users, in which several scenarios were demonstrated to users, which they then employed themselves in a real-world job interview scenario. This session also fed into the procedural/help documentation for Intek.

The study itself was then undertaken in which each user interviewed up to five randomly assigned interviewees and Intek was fully evaluated. Although the design was fixed at this point, only a handful of minor errors were logged. Some issues did arise, which are discussed in Chapter 5 Section 5.2.6.

This concludes the section on Intek user interface development. We now move on to describe the development of the back-end information extraction processes of Intek.

3.6 FACTOID INFORMATION EXTRACTION

This section describes the back-end processes that deliver Intek factoid content. These underlying extraction processes are the most complex elements of Intek, some marshalling state-of-the-art natural language processing techniques to filter information most pertinent to near-episodic test questioning. All IE processes eventually format their data for UI display, however this section focusses mainly on the back-end.

This section is broken down into five sub-sections that again represent stages in the development life cycle: we establish guidelines for selecting a set of topics and the data sources that will be used to extract information for each of these topics; we then apply these guidelines to an analysis of transcripts from a previous CCE study to

identify our initial set of topics and data sources; we examine good design guidelines for the extraction of information from web sources; we then describe our information extraction processes in detail; lastly, we give detail on the evaluation of these processes.

3.6.1 *Topic and data source selection top-down guidelines*

In this section we establish "top-down" IE guidelines and criteria for topic and data source selection, we then apply these criteria "bottom-up" in the next section by analysing transcripts from a previous CCE interviewing study to generate an initial set of topics and sources.

These IE guidelines are based on our main requirements. Each guideline is listed with the supporting high-level requirements from Section 3.5.1 in parentheses.

1. Undertake analysis to identify the topics and topic-based data sources most suitable for job interviewing. (1c, 1d)

2. Select data sources that can deliver formatted information in real-time. A few processes may be allowed to take longer, but the majority must return in real-time so the interviewer can begin questioning. (2a)

3. Select data sources that can deliver multiple facts from minimal inputs. For example, a job role search requires one piece of information, the name of the role, and might return multiple useful facts about that role, which is a good information input to output ratio. Conversely, an aeroplane flight search might require multiple pieces of information, the flight identifier and the time of departure, and return few useful facts that are not already on the ticket, which is a bad information input to output ratio. (2b)

4. Include some summary or background information data sources for each topic. Examples are a Wikipedia summary for a location or concept, or information summarised from the home page of an organisation web site. (4, 4a)

5. Select complimentary data sources for each topic that provide breadth and depth. Breadth refers to a selection of data sources that address different aspects of a topic, for example what an organisation does, its competitors, how you might travel there and people that work there. Breadth supports unexpected test

questioning and widens the fabrications required of deceivers. Depth refers to the obscurity of the information returned, for example the CEO of a large company will likely be general knowledge and not obscure enough for test questioning. Whereas the name of the person that created the web site will likely be too obscure (unless the interviewee coincidentally works in that team). By ensuring a range of data sources at different depths are available, at least some of the information returned by Intek should be useful no matter what the level of obscurity of the information snippet supplied for searches. Together, breadth and depth support unexpected tests of expected knowledge. (4b,3b, 3c)

6. Select data sources that contain factual information, to enable veracity-testable test questioning. A few sources that might contain questions only, such as "interview questions" regarding a job role, may be included to achieve additional breadth and depth. (3a)

7. Supply a good breadth and depth of data sources, but quality not quantity. All information needs to be parsed by the interviewer in real-time. Source selection should be careful and focussed. (3c)

8. Design for interviewee anonymity. No information will be supplied to the system that might uniquely identify an interviewee. Thereby, no social media or other data sources that rely on unique personal identification may be used. The main reasons behind this decision are increased complexity around GDPR legislation and security requirements, the added complexity and user interaction required in handling the ambiguity of interviewee names, and the difficult in recruiting for a non-anonymised study. (6a)

When selecting data sources, we aim for sources of facts that deliver the following three principles: good unexpected tests of expected knowledge; a complimentary range of facts; facts that minimise episodic distance. The next three sections describe these principles.

3.6.1.1 *Good unexpected tests of expected knowledge*

An unexpected test of expected knowledge is the ideal type of test question as defined by CCE.

An unexpected test refers to the selection of unpredictable aspects of a topic that would not be expected in a standard interview context, for example the "pitfalls" of an interest or the prices of equipment for that interest. This was described in our IE guidelines above as "breadth".

Expected knowledge refers to information an interviewee should know for a particular topic. This information should ideally be at an episodic or detailed-semantic level, similar to "depth" described in our IE guidelines. Also, expected knowledge questions should not be too easily fabricated, for example "what was the colour of that bus?" or pass-able, for example "what was the name of the barista?". Episodic and semantic knowledge was discussed in Section 3.1.4.

3.6.1.2 *Deliver a complimentary range of facts*

We have discussed the requirement to deliver a good breadth and depth of information. We expect the use of a variety of modalities of information will maximise the range of available test questions. By modalities of information we refer to the mixed use of textual summaries containing multiple facts in paragraph form, tabular lists of facts including text and images, bar graphs containing demographic information, question and answer pairs, images and points of interest. This variety when used well by an interviewer can make fabrications very difficult for deceivers, especially when visual, geographic, factual and social questions can all be combined for the same topic. For example, a visual question such "describe the company logo", a geographic question such as "describe your route to work", a factual question such as "tell me about your competitors" and a social question such as "remind me who is your COO?". It is not hard to imagine the pressure this combination of questions would put on a deceiver, while being relatively easy questions for a truth-teller.

3.6.1.3 *Minimise Episodic Distance*

Episodic distance is a theoretical guideline we have created to identify the distance of any concept or entity from an individual's episodic knowledge. We seek to minimise episodic distance to achieve a good level of "obscurity" from data sources in practice.

Episodic or detailed semantic knowledge is our aim in data source selection, however this level of detail is of no use if it is not specific to interviewee experience. We try to ensure this specificity to an individual's experience by selecting sources that are "episodically close"

to topic input snippets on one of several axes: geographic, for example local points of interest that are either very close or quite close and notable; people, for example those working in the same organisation or in the same team/location if for a large organisation; conceptually, for example common experiential questions asked about a job role, tool, or interest. The smaller the episodic distance, the more likely the information is to be relevant and expected knowledge.

Expected knowledge should be near-episodic or practically, detailed-semantic. In fact we aim for a balance between semantic general knowledge, which everyone knows, and obscurity, which no-one knows. The way we aim to filter our selection of possible data sources down to near-episodic, is by aiming to reduce semantic distance as far as possible. This can be done by cross-referencing multiple topic searches in an interview, and also during development by manually selecting sources that offer these types of entities and this level of information. For fixed data sources, for example constituency and MP information based on postcode, selecting a data source manually is straightforward as the level of obscurity of the postcode and of the constituency information will not change between searches. However, for dynamic sources, for example web site NER, we aim to balance the generality/obscurity of the input snippet against the generality/obscurity of each extracted entity or concept by providing multiple different sources at different levels of obscurity.

Topics themselves should be identified based on the potential duration of an interviewee's involvement with that topic, for example a home location or work organisation are clearly places at which an individual spends much time and as such should be aware of many facts regarding these entities. However, a travel trip, which was considered as a topic, is a fairly fleeting event which left the many possible questions easily deniable by the interviewee.

Clearly there is a risk that any data source might produce a fact that is not known to an individual interviewee, but use of these guidelines aims to maximise the likelihood of data source usefulness.

Having established a set of guidelines and principles for topic and data source selection, we now apply them through transcript analysis.

3.6.2 *Topic and data source selection bottom-up transcript analysis*

We apply our top-down guidelines to transcripts of a previous CCE interviewing study, in order to identify our initial range of topics

and data sources for extraction. We count all test questions asked in this previous study, grouping the questions into topics. Using our guidelines we then add any additional potentially useful test questions to the identified topics. We then score each question over five categories: technical feasibility (how achievable is the extraction of this question in real-time); estimated implementation time (is extraction achievable in a reasonible time frame); does the information support unexpected tests (not too predictable, not general knowledge); does the information support expected knowledge (relevant to episodic experience, near-episodic/detailed semantic, not too general or too obscure, not deniable/pass-able); is the information useful (does it fit the task of job interviewing well). Topics and data sources are then included based on the total scores of their constituent questions (higher is better) and the discrete pieces of input information required to extract those questions (lower is better). A good topic should also include the depth and breadth of potential questions, as discussed in the previous section.

Table 3.4 shows our final topic selection along with several topics that were rejected with explanatory reasons.

Having established our set of topics and data sources, we now move on to designing for extraction from the web.

Topic	Input Required	Topic Description	Development Status
Home	Postcode	A well-known location, usually Home, but could be any location travelled to.	Kept
Organisation	Organisation name	Various types of information and entities extracted from an organisation web site. Initially just a work organisation, but merged with study org, club and general points of interest.	Kept
Interest	Interest name	Specific facts, context and techniques around an interest, as well as local locations for pursuing that interest. Initially also covered occupation, but this was split out to Job.	Kept
Job	Job role name	Specific facts, context and techniques around an job occupation.	Kept
Tool	Tool name	Specific facts, context and techniques around a tool, especially a software tool or other device used in a job role.	Added for job interviewing
Course	Course name and qualification	Summarised information about a course, its location and key people.	Added for job interviewing
Study Organisation		A university or similar course provider.	Merged with Organisation
Point of Interest		A famous location or site someone might visit.	Merged with Organisation
Event		Dates, costs and other information regarding an event or gathering attended.	Merged with Organisation
Location		A general or low-duration location, for example a stopover.	Merged with Home
Interviewee		Personal information regarding an interviewee. Useful facts would need to be extracted from disambiguated personal web sites or social media, conflicting with the high-level requirement of anonymity.	Dropped
Home history		A history of places an interviewee has lived at, effectively linking multiple Home topics. Few useful facts available.	Dropped
Flight		Used frequently in airport security interviewing, not useful for job interviewing. Few available facts.	Dropped
Trip		Links multiple locations and modes of travel in a planned trip. Many inputs required for few available facts. Not useful for job interviewing.	Dropped
Related Person		Information related to a person being visited. Used frequently in airport security interviewing, not useful for job interviewing. Available facts either conflict with anonymity or are available from other topics.	Dropped

Table 3.4: Topics considered throughout the development of Intek. Input Required refers to the information snippet required to initiate a search. Development Status indicates a topic's eventual inclusion or exclusion.

3.6.3 *Information Extraction Design*

Design guidelines with respect to information extraction from the web refer to the sustainability of the practice, by protecting the web sites that are being accessed, as well as legal aspects, such as copyright and GDPR. Intek web-site-based IE follows web scraping best practice extracted from Zyte (2020) and described below.

1. Protect the web site. Limit the volume and frequency of requests so as not to burden web site server or interfere with normal operations. Intek web access is very low volume and frequency.

2. Inspect robots.txt. robots.txt is a plain text file stored in the root of most web sites, which gives instructions to compliant web scraping "robots" about which pages or directories should not be crawled. Intek inspects robots.txt.

3. Do not violate copyright. Original work contained within web sites may be subject to copyright and therefore storage and distribution of this work may be unlawful. Exception to copyright includes fair and transformative use, in which the original content is summarised or otherwise transformed, and factual content which is not generally covered by copyright. Intek focusses entirely on factual content and uses summary to condense paragraph-based textual information. Also, Intek does not store or distribute any information.

4. Do not violate GDPR. Do not store information that might identify an individual person. Intek does not access personal web sites or store any information.

5. Identify yourself. Use contact details in request headers to make yourself know to web site administrators.

6. Be aware of web site Terms and Conditions. If a login is required to access a site, the agreed terms and conditions must be adhered to. Intek does not use web sites that require a login, we make use of APIs where available.

Design choices must also be made around application performance. Near real-time response is an Intek requirement and the majority of sources conform to this, although exceptions are made for a very few longer running processes that return potentially key information.

In the case of these longer running processes especially and IE processes in general, wherever possible we make use of asynchronous execution rather than serially waiting for each process to complete. A performance-related choice is how to deal with Javascript-rendered web sites. These web sites are dynamically rendered after the initial base page has loaded, which may take a few seconds. It is possible to access information contained in these pages using a non-UI "headless" browser such as Google Chrome, but because of the initial few-second delay this method would cause to all web pages Intek accesses, and the relative infrequency of such pages being used, we choose not to implement this method, relying instead on the base page.

3.6.4 *Implementation and Evaluation*

As shown in Figure 3.26, Intek information extraction uses four main types of information source: API; corpus; fixed web site; dymamic web site. The next sections detail our IE approaches from these sources.

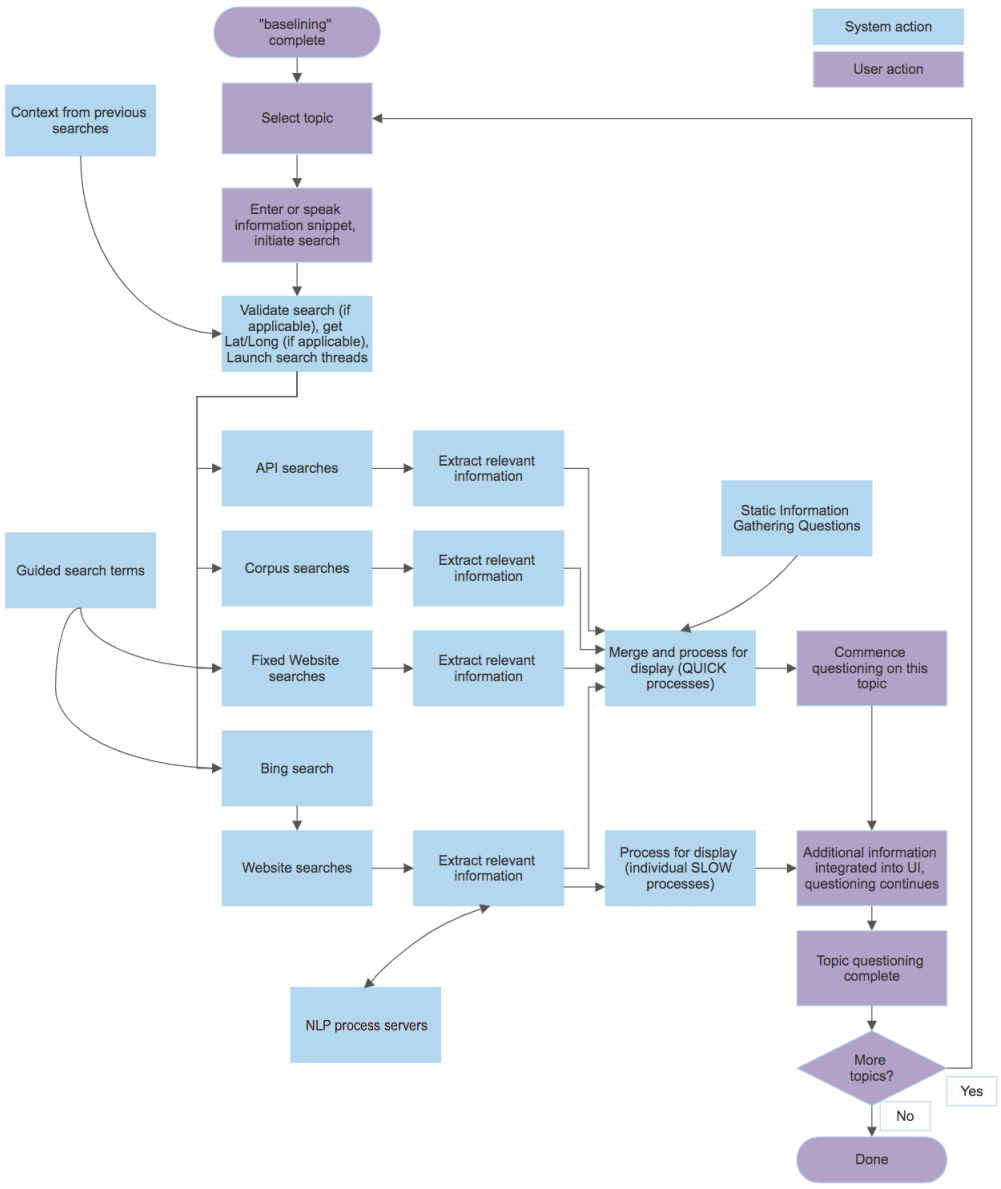

Figure 3.26 A high-level view of the Intek process and integration with CCE. The different data source types, asynchronous processing and UI integration are shown.

3.6.4.1 *API and Corpora*

API and corpora are the most straightforward Intek sources from both a development and extraction perspective. We select APIs that have been optimised for real-time retrieval of pertinent information around an entity, concept or location. Intek must then handle ambiguity around input snippets by informing the user where multiple similar alternatives are returned. Information is then transformed and formatted appropriately for display in the most efficient way, in terms of processing speed and UI ease of use. We use APIs to extract:

summary and Wikipedia "infobox" facts around entities and concepts from Wikipedia; to extract points of interest, travel routes between locations, pictures of organisations, reviews and opening hours and interests near a location from Google Maps; to extract web searches and web summary snippets from Bing Search; to extract key products, product categories, brands and prices from eBay.

We use Google Maps API as the canonical source for integration of other data sources for the Home and Organisation topics, as this API supplies a geographic location for a postcode or entity which other sources can use in turn to generate further information. If an entity is not present in Google Maps, as is often the case for smaller informal organisations, Intek tries to extract a postcode from the organisation web site via Bing search. This postcode is then used to extract an exact latitude and longtitude from Google Maps for nearby point of interest information.

APIs are used to provide cross-referenced information for various topics. Organisation uses the most recently searched Home location to generate a range of possible routes and modes of transport using Google Maps Directions API. Interest uses Bing search and Google Maps to generate local locations that might be used to carry out that interest.

Where information sources have been identified that appear to contain the correct level of information for Intek (see previous sections), but information retrieval is not optimised around the entity or concept of interest, or is too slow to be queried in real-time, where possible we download the corpus in question and perform optimisation offline for later real-time retrieval. We use this method to provide: information around Home location demographics from the UK Office for National Statistics at postcode level; UK Ofsted information on school locations and ratings; Times Higher Education World University Rankings information on student experience demographics; GeoNames information on nearby locations and population sizes.

3.6.4.2 *Fixed Web Sites*

Fixed web site sources are specific web sites, selected by hand, that contain facts that address our requirements, especially unexpected tests of expected knowledge. Development takes place initially by applying best practice to assess web site suitability, we then create wrappers which contain a set of rules which can be applied to a web site document object model to extract the desired pieces of information.

Once our wrappers have been created and tested, we then create a corresponding UI which is often developed and tested iteratively, with expert or user input. The final UI is then integrated into Intek or the source rejected if the information does not fit well into the interview flow.

Fixed web site sources used in Intek are: members.parliament.uk which is used by the Home topic to supply MP and constituency information; nationalcareers.service.gov.uk and other job sites that are used to supply summarised job role information for the Job topic; Google search which is used to supply frequently asked questions and answers from "people also ask" boxes for the Job, Tool and Interest topics; also from Google search, alternatives and competitors for Organisation, Job, Tool and Interest topics from the "people also search for" box. To extract other useful aspects of these topics, we use additional Google searches augmented with specific search terms. For example, the Job topic displays potential tools used in a role with an additional search for "<Job> + Tools", the Tool topic cross-references previous Job searches with the additional search "how is + <Tool> + used by + <Job>", the Interest topic displays practical information with the additional searches "<Interest> top tips" and "<Interest> common mistakes". Note that terms within "<>" are replaced by the specific information snippets supplied for those searches at run-time.

3.6.4.3 *Dynamic Web Site Extraction Using Bing Search API*

As entities and concepts supplied by interviewers in information snippets are variable and unpredictable, not all data sources can be predefined and selected in advance as with API, corpora and fixed web sites. Intek uses Bing Search API to dynamically provide web pages specific to interviewer searches. Different wrappers are then used to extract the desired information from the first Bing result. Certain Bing results are ignored if the source web site is unsuitable for extraction, for example YouTube, Facebook or PDF documents. Wrappers designed to extract from *any* web page that might be returned by Bing search must be more flexible in their design to ensure they consistenly return the desired information in the face of variable and unpredictable page design and content. These wrappers must also be thoroughly tested on a set of web pages representative of likely search results. Intek uses flexible wrappers in the following four scenarios, which we list in order of overall complexity, simplest first: heading and paragraph pairs, which extract paragraph elements then back-

track to identify headings; numbered lists, which extract itemised and enumerated lists in various formats; module information from course and degree web sites which can be in various formats; our named entity recognition pipeline which extracts names and locations from any page. We now discuss how these scenarios work in more detail, including the topics that use them.

The Organisation topic uses a simple generic wrapper to extract **heading and paragraph pairs** from the landing page of organisation web sites. This wrapper looks for any heading element containing more than two words followed by a single paragraph containing more than nine words with no child elements. This wrapper was designed to extract descriptive information from small-organisation web sites, particularly clubs, that may have no web presence other than a single page in a blog or larger site.

The Job, Tool and Interest topics use another wrapper to extract specifically formatted **numbered lists** from web sites. This wrapper looks for repetitive paragraph, list-item or heading elements that contain several words and are specifically formatted as a set of questions, an example is Q1. ...,Q2. ...,Qn. ..., but various formats are supported with a regular expression. At least five question items must be present for a list to qualify. This wrapper was designed to extract "interview questions" for Job and Tool topics, and "quiz questions" for the Interest topic. Such lists appear to be present on the internet for every conceivable role, technology and hobby. Additionally, this wrapper is used with the search term "<Interest> + top tips" to capture lists of this type.

The Course topic uses a more complex wrapper to extract **course module information** from an associated institution web site. Lists (ul, ol, dl tags) with headings that contain a keyword *module*, *structure*, *year*, *about the course* or *course detail* are extracted and displayed in the Intek UI whole, as long as the list does not contain other lists. Tables that contain at least two rows are also displayed whole, subject to a few exclusions. Headings (h1-h6 tags) followed by at least five paragraphs are extracted if each paragraph contains at least two words and the variance of word and element counts between these paragraphs is below a threshold. Groups of at least three "div"s are extracted if they each contain at least two words, have the same child element count and again the variance of word and element counts between divs is below a threshold. The aim of these rules is extract lists of course module information that have a consistent structure

and text content between elements, no matter which type of HTML tag is used to contain the information. Each set of extracted elements is returned to the Intek UI with an associated heading, which we locate by searching back up through the DOM tree for heading tags.

The remainder of this section describes our NER pipeline in detail, how it is integrated into the Intek UI and the underlying NLP technologies it makes use of.

The previously discussed wrappers target specific scenarios: lists of questions; headings with text; consistently structured text-containing lists of modules. Our **named entity recognition pipeline**, uses wrappers similar to those above to extract text elements, then applies a sequence of NLP named entity recognition (NER) tools which are used to identify and display text snippets containing mentions of real-world people and locations. NER tools are discussed in more detail in the Chapter 2 Section 2.4.

This NER pipeline is used by the Organisation topic to extract people of interest. This is particularly useful for clubs, societies and small or medium-sized companies where organisers or senior management are likely to be known to all members or employees, but will be unknown to the general public. The Course topic also uses this pipeline to extract people and locations of interest, which aims to identify specific buildings or facilities which genuine students should be aware of.

Figure 3.27 shows the overall entity extraction and presentation process. We now describe the steps in this process in detail.

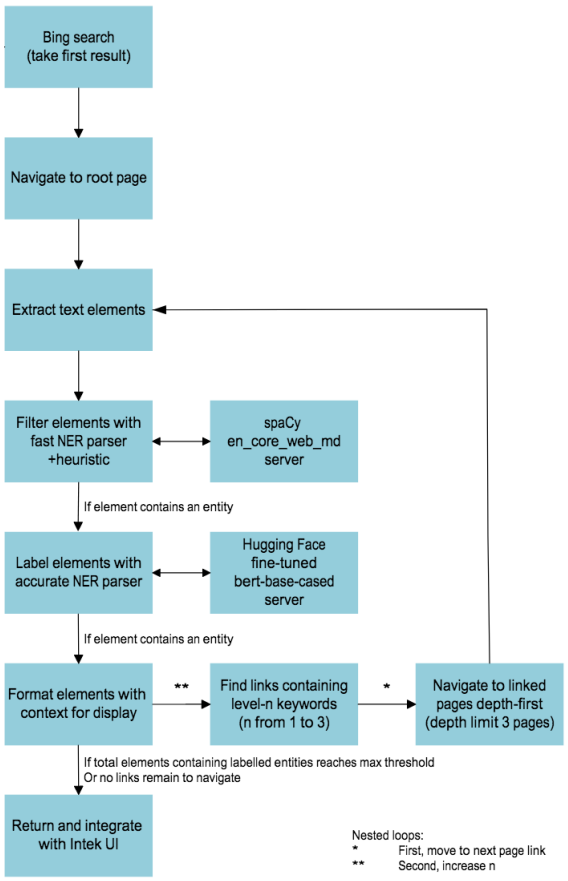

Figure 3.27 A high-level view of the entity extraction and presentation process used for Organisation and Course topics.

The process begins with a Bing search for the supplied Organisation or Course. The first result is taken with the aim of locating the main web page for the company, club, point of interest or course. For the Organisation topic, we then extract the root URL of the web site and assume this is the "home" page with which to begin site crawling, whereas the Course topic uses the URL supplied by Bing as this is likely to be the most relevant page to that course.

We then extract text elements from the initial page. This is done using wrappers similar to the previous approaches in this section. We extract potentially entity containing text areas contained anywhere in the page, not just the main content area. Areas like the header and footer may contain locations of interest, but we aim to exclude unwanted text-containing areas such as Twitter feeds, comments and navigation. We assessed content extraction or "boilerplate" removal technology for the extraction of these text areas, but we found they remove some areas of potential interest, such as locations in the page footer. Our wrappers extract all paragraphs, headings, table cells and

list items that contain text. We apply stricter extraction criteria for "div" and "span" elements as these are frequently nested and used for layout purposes, as well as containing text. We extract "div" or "span" groups of between two and five sibling elements that contain at least a few words, where each element has a consistent structure and contains few other elements. Our aim is to extract smaller groups of "div"s and "span"s which are used to contain text, not those used as style elements for presentation or as a high-level container for many other "div"s or "span"s.

Once we have extracted text elements, the next stage is to label entities of interest within this text. We do this in two stages shown in Figure 3.27 using different NER approaches, the first of which aims to filter out a potentially large volume of non-entity-containing text passages as quickly as possible, the second stage then applies the highest accuracy approach to labelling entities within the text.

For the first filtering stage, we use an "off-the-shelf" product spaCy (Honnibal and Montani, 2017), which with version 2.0.18 has good accuracy using a CNN-based neural model, but is especially designed for speed at 10,000 words per second. spaCy is focussed on accomplishing "basic" NLP tasks such as part-of-speech labelling, sentence dependency parsing and NER, as opposed to more general language understanding and generation. This approach allows the spaCy architecture to be relatively lightweight with fewer parameters and input dimensions, which contributes to its state-of-the-art speed. spaCy is typically used as a preparation tool for further processing and indeed spaCy themselves claim spaCy is "the best way to prepare text for deep learning".

As with most modern NLP approaches, spaCy relies on a pretrained model as a starting point for labelling or further training. We use the medium English model *en_core_web_md* which is a good balance between size, speed and accuracy. This model is trained on the GloVe Common Crawl and Ontonotes v5.0 datasets and contains non-contextual word vectors, which allow a statistical measure of similarity between words, by looking at which words occur close to other words over a large quantity of training data. The model also contains binary weights extracted from the neural network after training, which allows the model to predict labels, such as named entities, using word vectors in the context of the supplied sentence. For named entity recognition, the model is trained to label 18 entity types, including PERSON, GPE (a geo-political entity location) and ORG which

Intek uses. The medium model, although relatively small at 115Mb, does carry an overhead in terms of loading and startup time. For this reason we run our spaCy instance "spun up" as a WebSocket server, to give a very quick response in terms of latency.

Intek then keeps only potential entity-containing elements using the presence of a labelled entity of type PERSON, GPE or ORG from spaCy. In order to catch any further entities that spaCy may have missed, Intek also checks for the presence of standard titles (Mr, Mrs etc.) that might indicate a name. Also, any GPEs labelled must occur in the same text passage as a day of the week, which limits the typically quite frequent GPE mentions to only those that indicate a meeting place. Results from Table 3.5 show that, on average, this spaCy stage is working as expected, by filtering out a large percentage of extracted page elements, paticularly from pages we would not expect to contain names of people, for example the root page of a larger company web site.

For the second filtering stage, the final high-accuracy NER labelling of our remaining set of text elements is done using a pre-trained BERT (Devlin et al., 2018a) English bert-base-cased model. This model has been fine-tuned on the CoNLL-2003 NER dataset (Tjong Kim Sang and De Meulder, 2003) to recognise four coarse entity types: PER (persons), ORG (organisations), LOC (locations) and MISC (other). This approach achieves state-of-the-art accuracy on the CoNLL-2003 shared task, which is a standard benchmark in NER. The excellent accuracy of BERT is due in part to its large stacked encoder architecture, which in the case of the bert-base model, consists of 110 million parameters made up of 12 encoder layers each containing 12 self-attention heads. This network is pre-trained on 800 million words from the BooksCorpus and 2.5 billion words from English Wikipedia. This is considerably less training data than spaCy, but BERT uses only documents that have a good sequential sentence structure. BERT uses two tasks for pre-training. One of these tasks is masked language modelling (MLM), in which words in input sequences are randomly masked or hidden and must then be predicted by the network using information from the whole sentence. In addition a next sentence prediction (NSP) task is performed concurrently, in which the network must predict the order of two input sentences. MLM helps the network model context within sentences, while NSP deals with context between sentences. The last key part of BERT is self-attention which identifies important relationships between words in sentences, thereby gaining

some "understanding" of that sentence. Self-attention uses a matrix of similarity between each word in an input sequence, which is then combined over different semantic aspects of the sentence to produce context-aware word representations, which can be passed to the feed-forward layers in the network. Once a BERT model has been pre-trained, which requires considerable resources for a network of this size, it could be said that the network has a good "understanding" of language. The network is now deemed a "language model" and is suitable for a number of downstream tasks, including named entity recognition.

	Average element count			
Entities expected in page	Extracted from page	Filtered out by spaCy	Filtered out by BERT	Remaining
N	95.7	93.1 (93%)	1.6 (63%)	1.0
Y	161.6	122.6 (61%)	4.2 (22%)	34.8

Table 3.5: Results from the NER pipline. Results are broken down by whether the page is expected to contain a list of people, for example "meet the team" pages. Results shown are the average of initial text elements extracted from web pages, the number and % of elements removed by the spaCy filter, the number and % of remaining elements removed by the BERT filter and the number of remaining text elements labelled with one or more entities.

As with the spaCy model, this fine-tuned BERT language model is quite large and takes significant time to start up, so we devote another WebSocket server to maintaining a running BERT instance.

To filter text elements for final display, we keep only those elements containing a PER label or a LOC and a day of the week. Text elements are then formatted into "snippets" by including a text window of 20 characters around a labelled entity. Snippets are then grouped by entity to reduce duplication. Potential roles, such as CEO, are highlighted for the interviewer by matching against a lookup list. Snippets formatted in this way allow the interviewer to scan a whole list of people by name as quickly as possible, but also quickly identify the context of that person within the organistion by checking their role. The whole extracted passage can be found by clicking on each snippet. Figure 3.28 shows a fully-populated people factoid. As seen in Table 3.5, the BERT model is making some additional exclusions over spaCy. Testing indicates BERT gives more consistent entity boundaries as can be seen in Figure 3.28.

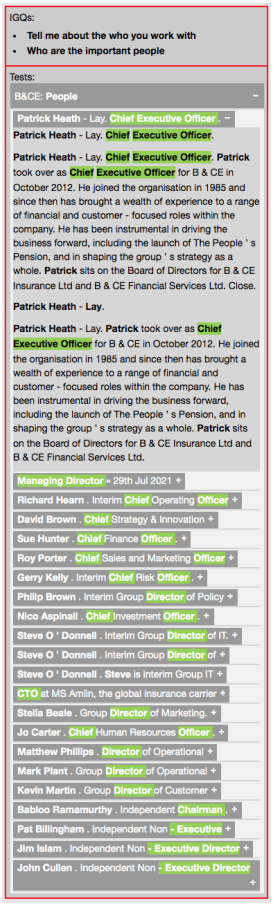

Figure 3.28 A People factoid containing a list of key staff extracted from a company web site. Names are highlighted with bold text, associated roles are highlighted in green. The first name has been opened to reveal the full grouped text elements extracted containing that entity.

The NER pipeline process shown in Figure 3.27 iterates by following links from the initial page deeper into the web site. This behaviour is essential, as especially for larger organisations, pages containing person names of interest are likely buried several page-layers deep into the site. For practicality, Intek looks only 3-pages-deep and only examines pages with titles containing words in a lookup list, for example "team" or "staff". The NER pipeline process continues to extract entities from pages, as detailed above, until either the process runs out of pages to examine or the number of snippets accumulated reaches a threshold.

We have now detailed the NER pipeline process and its underlying technology. We now describe one final WebSocket server used for summarisation, before moving on to IE evaluation.

3.6.4.4 *WebSocket Servers*

We have mentioned the two WebSocket servers that support the NER process. One further server is used by all topics to summarise some chunks of text, especially the initial summaries for each topic. This extra summarisation gives the interviewer less text to parse before getting to the salient points of the summary. We use BART (Lewis et al., 2019) for abstractive summarisation, which involves the interpretation and re-wording of the original text. BART is an autoencoder transformer which achieves state-of-the-art results on the X-Sum abstractive summarisation dataset. Intek uses the Wikipedia API to identify a relevant summary. This summary is then passed to the BART WebSocket server for further summarisation and abstractive reduction. The final summary should include the key facts of the original with irrelevant detail removed.

3.6.5 *Evaluation*

Evaluation of new information extraction sources was performed initially in "unit testing". This applied wrappers and NLP technology to a range of representative web sites, which allowed robust backend processes to be developed. Once IE processes were functioning adequately, prototype user interfaces could be created to enable stakeholder evaluation of functionality. We used standalone prototypes in the early days, which could be quickly iterated as a result of feedback and refined into feasible methods for detection deception. As Intek developement progressed, these prototypes were integrated into a coherent single Intek user interface and testing became more concrete, using CV-based testing. The number of CVs used for testing increased throughout development, which as a result of feedback, made IE processes more robust and identified further processes to be implemented. Finally, end-user testing was made available which allowed Intek to be honed for novice-interviewer users.

3.7 CHAPTER SUMMARY

This chapter presented the entire lifecycle of designing and implementing Intek. We gave an overview of the CCE method on which Intek is based, then described how the key elements of CCE relate to Intek. This gave an insight into the motivations behind Intek design.

We then described the iterative development life cycle we used. We discussed key high-level design decisions that shaped Intek to a large extent. We then gave detail on every step in the lifecycle as it progressed for development of the front-end user interface and the back-end information extraction processes.

This chapter gives full context for next two chapters, in which we discuss the development of a web-specific NER technique (Chapter 4) and the evaluation of Intek in our interviewing study (Chapter 5).

4

LEVERAGING HTML IN FREE TEXT WEB NAMED ENTITY RECOGNITION

In this chapter, we present a novel method for extracting entities from web pages. This method is related to our work with Intek in Chapter 3, specifically the extraction of person names and locations from organisation web sites. This chapter is split into three sections. We begin with an introduction to our approach, its motivations and contribution. We then describe our experimental approach, including details of the datasets, models and evaluation used. We then present our results and analysis for this work.

4.1 INTRODUCTION

In this introductory section, we first give a summary of the whole chapter, we then explain our motivations for undertaking this work and finally list its contributions.

4.1.1 *Summary*

A summary of this chapter is as follows. HTML tags are typically discarded in free-text named entity recognition from web pages. We investigate whether these discarded tags might be used to improve NER performance. We compare Text+Tags sentences with their Text-Only equivalents, over five datasets, two free-text segmentation granularities and two NER models. We find an increased F1 performance for Text+Tags of between 0.9% and 13.2% over all datasets, variants and models. This performance increase, over datasets of varying entity types, HTML density and construction quality, indicates our method is flexible and adaptable. These findings imply that a similar technique might be of use in other web-aware natural language processing (NLP) tasks, including the enrichment of deep language models.

4.1.2 *Background and Motivation*

Named entity recognition (NER) is the identification of the proper names of objects. NER can be informational in its own right, but also serves as pre-processing for other information extraction (IE) tasks. The web as a data source for NER offers opportunities as a massive corpus of structured information and free-text, but also presents challenges of a far from uniform layout and variable writing style; the web is formatted for ease of reading, not for ease of extraction.

There are two main methods for NER on the web: wrapper methods, that perform well with HTML when extracting particular structured elements from web pages, but do not have the flexibility to exploit the variable language of free-text sentences; NLP NER methods represent the state-of-the-art in entity extraction from unstructured free-text sentences, but do not exploit HTML formatting at all. The next two sections give background in these two areas.

4.1.2.1 *Structured elements and wrapper methods*

Structured text is contained within records, lists (semi-structured elements) and tables (structured elements). Entity extraction for these elements is usually performed with wrapper methods that identify text areas using precise contextual patterns. HTML is an essential delimiter of records in Wrapper methods. Techniques for handling HTML tags in web pages range from DOM tree matching (Chang and Lui, 2001; Crescenzi et al., 2001; Qiujun, 2010) to convolutional neural network based visual approaches (Gogar et al., 2016). HTML tables have had specific approaches applied (Gatterbauer et al., 2007; Cafarella et al., 2008; Dalvi et al., 2012). Wrappers are insufficient for dealing directly with the variability and ambiguity of free-text.

4.1.2.2 *Unstructured text elements and NLP NER methods*

Unstructured free-text natural language elements, such as paragraphs "<p>", make up the majority of web content. Most recent free-text NER approaches make use of sentence-based NLP neural network techniques, such as LSTM+CRF Ma and Hovy (2016) or more recently pre-trained language models, such as BERT Devlin et al. (2018a). NLP techniques that use the web as a source typically discard any HTML contained in these free-text sentences. The reasons for discarding HTML seem to be expectation, convenience and heritage. HTML in

natural language is not expected to be of any benefit: "free-text exhibits no implicit structure at all" (Goebel and Ceresna, 2009). Any information to be extracted from free-text is assumed to be held entirely in natural language grammar and semantics. Many NLP code libraries, such as Beautiful Soup (Richardson, 2007), strip HTML with a single function call: "Since so much text on the web is in HTML format, we will also see how to dispense with markup." (Bird et al., 2009). A plain text approach to NER is used in various genres of web site, from web newspapers (Ekbal et al., 2012; Wibawa and Purwarianti, 2016), the social web (Russell, 2013) and web sites identified through web search (Tu et al., 2005; Bunescu, 2007; Speck and Ngomo, 2014) to elements of well-researched shared tasks, such as the scientific web page section of BioNLP 2013 (Nédellec et al., 2013), weblog sections of CoNLL-2003 (Sang and De Meulder, 2003) and Ontonotes v5.0 (Weischedel et al., 2013).

In the five datasets used in this chapter, HTML tags make up between 10% and 34% of tokens, when using one token per tag and a separate token for open and close tags.

The case for the inclusion of HTML in free-text NER is unclear. Opposing possibilities are that words should maintain their integrity, for example "Start" should be rendered as "Start", and thereby the inclusion of these HTML tags would negatively affect performance. Conversely, in the tokenized sequence "<p> A merry fellow is <a> Santa Claus ...", recognition of the entity Santa Claus may benefit from the direct delimit of the "<a>..." element. The main question we ask in this chapter is, what effect does the inclusion of HTML in NER have on performance? We also consider the reasons for any performance differential and which types of web page might benefit most. Also, can HTML be included in current NLP practices with minimal additional labour or processing effort? The need for this enquiry is reinforced in an IE review (Small and Medsker, 2014) who state "traditional methods for linguistic analysis do not exploit the tags and layout formats implicit in online text" and "adaptive methods are needed to handle the variety of types of text".

HTML tags have been used in NER previously, seemingly as a side effect of a combined approach to NER from both structured and free-text elements (Soderland, 1999; Whitelaw et al., 2008; Mirończuk, 2018). We observe that these approaches typically employ the same split of structured and free-text techniques detailed above. To our

knowledge, these are the only approaches that make use of HTML tags in free-text areas.

4.1.3 *Contribution*

In this chapter we seek to answer the following questions, which form our contribution:

- To what extent does the inclusion of HTML tags in free-text affect NER performance?
- What are the causes of this effect and which web pages benefit the most?
- Can HTML tags be included efficiently in sentence based NER?

These contributions are discussed in more detail in our conclusions Chapter 6 Section 6.2. We now explain our experimental approach.

4.2 APPROACH

We conduct NER experiments on the five datasets summarised in Table 4.1 and detailed below. Three of these datasets include "gold standard" entity annotations, which are compared against web pages in order to label entities. Some of these datasets include a collection of web pages to be labelled, while others require their extraction from the web. The remaining two datasets are labelled using distant supervision, which uses entity mentions from DBpedia to label web pages extracted using Bing search.

We split each dataset into two variants by extracting two sets of free-text elements from each, a paragraphs set and a paragraphs plus headings set. This enables us to explore the effects of different types of sentence. We extract the inner HTML contents from the <body> of raw web pages, removing script and comment blocks. "Set1" then contains <p> tags and contents, "Set2" contains <p> and <h...> tags and contents. We ignore any structured elements contained in a table record, list item or option, as we are only concerned with *free* text. The contents of these elements are then tokenized on whitespace and HTML tags, including *all* tags contained within. Each tag is simplified into a single token, removing attributes and style. These free-text sequences are then sentence segmented using NLTK Bird et al. (2009) to create the Text+Tags variant. HTML is stripped from each sentence

to create a directly comparable Text-Only variant. "<h3> <a> Australia and the world </h3>" is a typical Text+Tags sentence.

4.2.1 *Datasets*

We use five datasets in these experiments: RE3D, SWDE, WEIR, Persons and OrgPersons. RE3D is assembled by applying a supplied "gold standard" set of annotations to live web pages using the links provided. The fact that live web pages are used means there is a small chance that the extracted content of RE3D might differ between this and other research using this dataset. SWDE and WEIR are assembled by applying a supplied gold standard to web pages that are also supplied with the dataset, therefore SWDE and WEIR are immutable. RE3D, SWDE and WEIR are publicly available. Persons and OrgPersons were generated by us using distant supervision from DBpedia based on Mintz et al. (2009).

These datasets provide a contrast of data generation methods, genres, entity types, tag densities and evaluation mechanisms. Especially, differences in dataset size and volume of sentence duplication have led us to use slightly different training approaches for each dataset. These approaches are detailed below, but it should be noted that the same settings are always used for Text-Only and Text+Tags variants within the same tagset, to ensure a direct comparison.

	Entity		Sentence count		Tag density%		Avg. sentence len.		
Dataset	Types	Categories	Set1	Set2	Set1	Set2	Set1	Set2	Construction
OrgPersons	1	1	8,198	10,901	11	13	24.6	20.1	Distant
Persons	1	1	121,598	186,523	16	21	22.4	15.9	Distant
RE3D	14	7	1,393	2,528	11	19	15.6	11.3	Gold
SWDE	16	8	49,849	113,902	10	12	22.0	19.4	Gold
WEIR	14	4	3,796	10,858	30	34	22.6	6.4	Gold

Table 4.1: Dataset and tagset summary. Tag density% shows HTML tag tokens as a proportion of all tokens. Average sentence length includes HTML tag tokens. Construction indicates whether the dataset was labelled using included gold standard annotations or distant supervsion from DBpedia.

RE3D[1] (Science and Technology Laboratory, 2017) created on behalf of the UK Defence Science and Technology Laboratory, contains

1 https://github.com/dstl/re3d

entities relevant to the role of a defence and security intelligence analyst. We generated this dataset from the live web pages of seven sites using the supplied gold standard. The gold standard was created using a hybrid of automated extraction, expert annotation and crowdsourcing for web pages from the Australian Department of Foreign Affairs, BBC Online, CENTCOM, Delegation of the European Union to Syria, UK Government, US State Department and Wikipedia. Entity types included are CommsIdentifier, DocumentReference, Frequency, Location, MilitaryPlatform, Money, Nationality, Organisation, Person, Quantity, Temporal, Url, Vehicle, Weapon. We evaluated this set with 6 x 5-fold stratified cross validation. Dataset-specific settings are used in training for LSTM: 200 hidden layer units, 200 epochs, and BERT: 20 fine-tuning epochs. LSTM and BERT are trained only on sentences containing at least one labelled entity.

SWDE[2] (Hao et al., 2011) contains entities from eight semantically diverse categories for structured web data extraction testing. We generated this dataset from the supplied cached pages using the supplied gold standard. This dataset contains 124K cached web pages from eighty web sites from eight semantically diverse categories: Autos, Books, Cameras, Jobs, Movies, NBA Players, Restaurants and Universities. Each category contains a fixed set of three to five entity attributes, for example, Autos contains model, price, engine and fuel-economy. The gold-standard was generated for these pages by applying "carefully prepared" handcrafted regular expressions. These expressions and entity type sets were created through observation of ten popular sites in each category. This larger set is evaluated with 5 x 2-fold stratified cross validation. Dataset-specific settings are used in training for LSTM: 200 units, 10 epochs, CuDNNLSTM instead of full LSTM, and BERT: 10 epochs. Keras CuDNNLSTM lacks recurrent dropout, but makes better use of GPU resources than the standard LSTM. LSTM is trained over the whole dataset including sentences containing no labelled entity. BERT uses a 1:5 ratio of entity to non-entity containing sentences for tag Set1 and used the whole dataset for Set2.

WEIR[3] (Bronzi et al., 2013) was constructed to test web extraction and integration of redundant data, by creating rules to identify data structures overlapping multiple pages and extracting the least redundant. The dataset contains entities from 24k web pages from forty web sites over four categories. Categories are soccer players,

2 https://archive.codeplex.com/?p=swde
3 http://www.dia.uniroma3.it/db/weir/

stock quotes, video games, and books. Each category contains a fixed set of four to nine entity attributes, for example, video games contains publisher, developer, ESRB rating and genre. The gold-standard annotations were generated by manually composing extraction rules, then mapping these to a set of pages extracted by a combination of set expansion using web search and querying selected finance and bookstore sites. We generated this dataset from cached web pages supplied with the dataset by applying the gold standard. We include an additional text containing tag, anchor <a>, in tag Set2 due to a sparseness of <h...> tags in this dataset. This dataset is evaluated with 6 x 5-fold stratified cross validation. Dataset-specific settings are used in training for LSTM: 200 units, 20 epochs, batch size 128, and BERT: 10 epochs. LSTM is trained over the whole dataset including non-entity sentences. BERT uses a de-duplicated dataset for tag Set1 and a 1:5 ratio of entity to non-entity sentences for tag Set2.

Persons was constructed to test distant (weak) supervision Mintz et al. (2009), for the task of extracting person attributes from organisation web sites. This set is the first stage of this process and contains only person name entity types. The construction process extracts all persons from DBpedia (Bizer et al., 2009), then uses the top-10 web search results for each person name, minus some exclusions such as YouTube, as a page corpus. Each page has free-text areas extracted, then direct string matches for forename and surname, plus matches including possessive apostrophes, are labelled. We use a headless Google Chrome browser to render pages before extraction, in order to capture JavaScript rendered pages as well as standard static pages. Any interactive pages that are encountered are treated as a static page. Distant supervision is a noisy process (Roth et al., 2013); we exclude noisy sentences by the presence of a non-labelled forename or title present from two lookup lists. These exclusions reduce the likelihood of unlabelled names. The forename exclusion list contains the names extracted from DBpedia, the title exclusion list is manually created. From 143k DBpedia persons, we extract 122k sentences containing an entity using Set1 and 187k sentences from Set2. This set is evaluated against a hand labelled set of 1,214 sentences extracted from thirty web sites. Annotation of our evaluation set was performed by the authors, with an inter-annotator agreement of 98.5%. Disagreements were due to incorrectly labelled titles, these were corrected so titles were not labelled. Dataset-specific settings are used in training for LSTM: 100 units, batch size 64, 50 epochs. LSTM is trained on

entity containing sentences only. BERT uses a 1:0.2 ratio of entity to non-entity sentences.

OrgPersons, constructed in a similar way to Persons, still contains only person entities, but is oriented toward a person's employing organisation, rather than the person themselves. Organisations and corresponding key persons are extracted from DBpedia, with the top-5 web search results from the organisation name processed. Each result is processed three pages deep by following same-domain links depth-first, with the same matching and exclusion criteria as Persons. We perform this link-following in order to locate and label mentions of a person that occur deeper in the web site navigation hierarchy. From 11k DBpedia company/keyPerson relations, 8k sentences are extracted for tag Set1 and 11k entity containing sentences for tag Set2. This set is evaluated using the same hand-labelled test set as Persons. This is a smaller task-focussed set; the three page deep processing is likely to hit more of the types of pages that were labelled in our evaluation set. Dataset-specific settings are used in training for LSTM: 100 units, batch size 64, 50 epochs. LSTM is trained on entity containing sentences only. BERT used a 1:1 ratio of entity to non-entity containing sentences.

4.2.2 *Models*

We evaluate sentences using two NER models that have previously achieved state-of-the-art results on the CoNLL-2003 NER task (Sang and De Meulder, 2003). Our first model is based on Ma and Hovy (2016), which combines a character-level convolution and word-level embeddings into a single representation for each word. Word-level embeddings are trained to represent the contextual meanings of each word. We use two separate sets of word embeddings for comparison, one Word2Vec skip-gram set (Mikolov et al., 2013) trained over our five datasets and one Stanford GloVe set (Pennington et al., 2014) pre-trained over Wikipedia 2014 and Gigaword 5. This sequence of concatenated word representations is fed into a bi-directional LSTM, then a CRF layer for sequence labelling. The second model, known as BERT (Devlin et al., 2018b), is based on a transformer architecture in which the BERT-base-cased model uses twelve encoder layers each with twelve attention and feed-forward layers including 110M parameters in total. Input to the encoder is handled through sub-word tokens known as WordPieces. BERT generates a deep language rep-

resentation pre-trained on a language modelling task over the BooksCorpus and English Wikipedia. This general language model can be fine-tuned on many NLP tasks including NER. We use the Hugging Face (Wolf et al., 2019) PyTorch BERT-base-cased implementation for BERT fine-tuning, which provides a WordPiece tokenizer and adds a linear CRF layer language model for sequence labelling.

Our word embeddings are trained over all our five datasets combined. They have the same variants we use for the main datasets: paragraph, paragraph plus heading, text-only and text plus tags, and are used in conjunction with their respective dataset variants. This is a fairly small corpus, so we include the pre-trained GloVe word embeddings as a baseline for the text-only approach. All our embeddings are created using Gensim Word2Vec skip-gram (Rehurek and Sojka, 2010) with 100 dimensions, twenty iterations and a window of five. This configuration was optimised on the text-only paragraph variant.

These models and embeddings provide a good contrast for our experiments.

Hyperparameter settings used by all datasets are as follows. LSTM uses Keras/Tensorflow (Chollet et al., 2015) CNN/Bi-LSTM/CRF with a batch size of 64, dropout of 0.5, recurrent dropout of 0.5, crf_loss loss function and Adam optimiser with an initial learning rate of 1e-3. BERT uses a batch size of 16 and Adam optimizer, with an initial learning rate of 3e-5. Dataset-specific configurations are listed above; these have been optimised using random search.

4.2.3 *Evaluation*

We use the exact match metric as used by CoNLL-2003 Sang and De Meulder (2003), which requires a predicted label to match the exact same words as in the evaluation set.

Tag density% is the percentage of all tokens that represent HTML tags. Average sentence length is calculated over tokens in Text+Tags variants.

4.3 RESULTS AND ANALYSIS

Our results in Table 4.2 show increased F1 performance for Text+Tags over every dataset, tagset and model.

	Text-Only LSTM		Text-Only	Text+Tags LSTM		Text+Tags	Text+Tags
Dataset.Tagset	GloVe	W2V	BERT	GloVe	W2V	BERT	Improvement
OrgPersons.1	85.5	84.0	80.5		86.1	**88.1**	+2.6
OrgPersons.2	85.1	82.7	81.2		**89.0**	87.1	+3.9
Persons.1	70.9	67.3	69.0	**73.8**	70.3	72.8	+2.9
Persons.2	74.1	70.6	70.3	**77.2**	71.0	76.8	+3.1
RE3D.1	72.4	71.6	71.9		72.8	**73.4**	+1.0
RE3D.2	74.7	74.3	73.6		**75.6**	75.0	+0.9
SWDE.1	61.7	64.0	74.8		**76.6**	76.0	+1.8
SWDE.2	66.8	68.0	82.6		84.6	**86.6**	+4.0
WEIR.1	75.4	72.9	87.5		89.4	**91.5**	+4.0
WEIR.2	64.1	65.2	70.5		**83.7**	73.4	+13.2

Table 4.2: Full F1 results. The improvement of the best Text+Tags approach over the best Text-Only approach is shown.

We now discuss the main elements of the experiment that we believe has influenced these results: sentence characteristics, entity delimitation and models.

4.3.1 *Sentence Characteristics*

Table 4.2 shows the F1 improvement of the best Text+Tags approach over the best Text-Only approach. This improvement correlates fairly well (Pearson correlation coefficient of 0.72) with the tag densities in Table 4.1, suggesting dataset tag density is applicable for assessing a future dataset for our Text+Tags technique. We found a weaker (-0.56) negative correlation between sentence length and performance, indicating shorter sentences perform somewhat better. However, dataset characteristics are more complex than those general statistics listed in Table 4.1. Analysis of the distribution of sentence tag densities, reveals two main types of sentence: natural language sentences containing some tags and repetitive tag dense patterns. Natural language sentence tag density peaks at zero (plain text), with sentence quantity decreasing toward a tag density of around 40%. Specific tag dense "sentences" appear in large quantities around specific higher densities, indicating repetitions of the same semi-structured patterns. These patterns still contain natural language or entities, but are often missing punctuation, relying on HTML tags to delimit. A good example of this is "<h1> John Smith </h1>", where this pattern is repeated

many times in our Set2 variants, giving a large quantity of 50% tag densities. Most dataset variants contain overwhelmingly natural language. OrgPersons.2, Persons.2 and RE3D.2 contain natural language mixed with large density spikes at 50% indicating the heading name as above. WEIR.2 contains little natural language with four big spikes at different densities indicating four different repetitive tag patterns. We find Text+Tags performs slightly better for variants that contain a mix of sentence types and performs much better for WEIR.2 which is pattern dominant. Analysis also shows the variability of automatic annotation from gold standard, with SWDE in particular missing many labels. The consistent Text+Tags performance over these variables demonstrates the adaptability of this approach.

4.3.2 *Entity Delimitation*

We analysed LSTM results, looking at ratios between occurrences of tags that delimit successful and unsuccessful entity labels. We find that across our different datasets, HTML tags delimit between 16% and 31% of entities and that entity-closing tags have a entity-labelling success ratio between 122% and 251% better than entity-opening tags. An example is the sentence "<h2> Traveler Advice on Little Delhi Restaurant </h2>" where the labelled entity is Little Delhi Restaurant. Where we look at success ratios for individual HTML tags, we find opening tags , <h...>, , , <a>,
, , and closing tags
, , , </h...>, , <img>, , , perform well, while opening tags , ,
 and closing tags ,
, perform poorly. These lists are ordered by best/worse performance and are not exhaustive. Interesting points are that close tags are good openers and vice versa, the variable performance of
 as an open or close tag, and that italics performs much worse than other formatting options, perhaps indicating poor annotation quality in SWDE where this was especially prevalent. Poor performing tags might be pre-processed out in future work to further improve performance.

Good performers
Opening tags: , <h...>, , , <a>, , ,
Closing tags: , , , </h...>, , <img>, , ,
Poor performers
Opening tags: , ,
Closing tags: , ,
Entity-closing tags success ratio is between 122% and 251% better than entity-openers

Table 4.3: Entity delimit tag performance

Models

BERT shows a stable Text+Tags improvement between dataset variants, where sentence tag density can vary considerably and between datasets of differing quality. We observe increased precision on the single entity datasets: OrgPersons and Persons, with increased recall for other datasets. BERT is able to adapt to Text-Only patterns in the WEIR dataset that the LSTM model fails on. Our LSTM Word2Vec Text+Tags outperforms BERT 4/10 times, which might indicate a deep language model trained on web text including HTML tags may further improve performance. Figures 4.1 and 4.2 show precision and recall scores for all datasets and tagsets.

Our own LSTM Word2Vec embeddings show a Text+Tags improvement over all datasets and variants. On the Persons dataset, GloVe embeddings outperform our Text+Tags embeddings. This prompted us to experiment with Text+Tags GloVe which unexpectedly scored best overall for this dataset. This Text+Tags GloVe performance may indicate our own embeddings would benefit from more extensive training in both volume and context window.

LSTM shows a large Text+Tags improvement for SWDE and WEIR datasets due to poor precision on Text-Only. For the WEIR dataset, this is mainly due to patterns that are almost un-differentiable without delimiting tags. For SWDE, this is due to poor annotation from the gold standard, introducing many false negatives. An example is the sentence "What initially grabs your attention in The Five People You Meet in Heaven ?", the book title in bold is recognised with the tags present, but not without. Text+Tags and the more complex BERT models can deal with these scenarios.

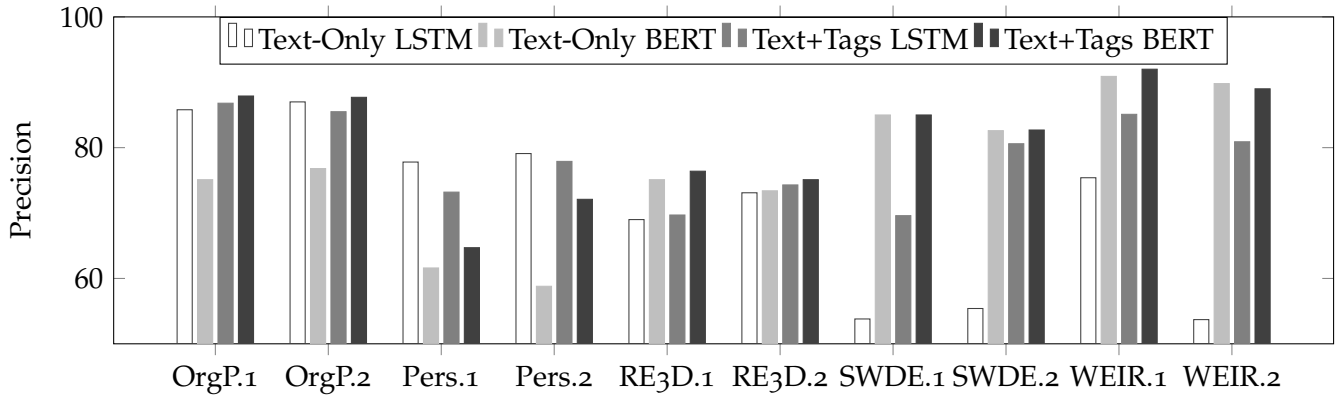

Figure 4.1 Precision results for each dataset, tagset and model. BERT performs worse for OrgPersons and Persons, while LSTM performs worse for SWDE and WEIR with a large differential between LSTM Text-Only and Text+Tags.

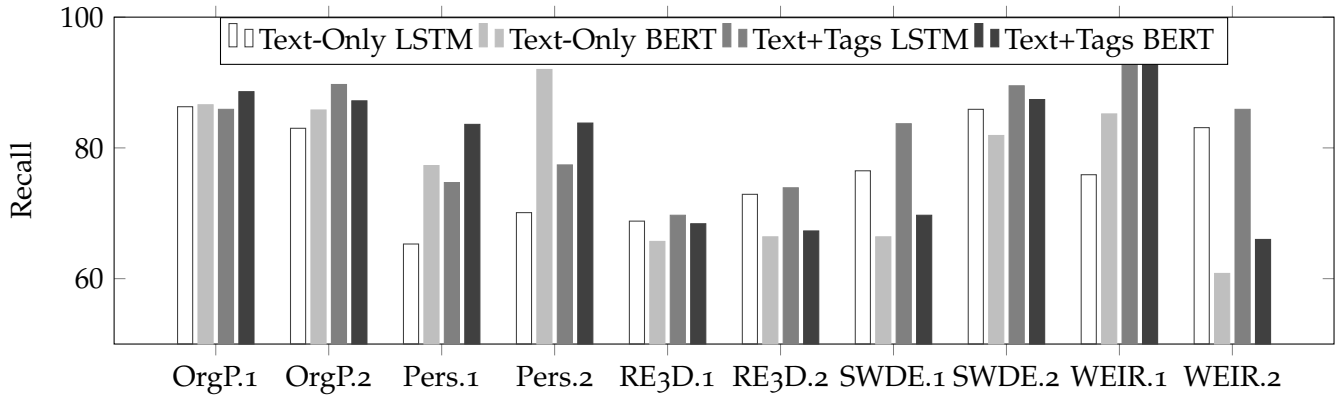

Figure 4.2 Recall results for each dataset, tagset and model. LSTM performs worse for Persons, while BERT performs worse for SWDE and WEIR.2, reversing the trend from precision.

4.4 CHAPTER SUMMARY

This chapter has detailed our experiments in applying NER to the web. We find that our method performs well, is flexible and is relatively easily implemented. This approach is applicable for use in the Intek NER pipeline. However, in further experiments we find that BERT fine-tuned on a larger hand-labelled dataset *without* HTML included (CoNLL 2003, 14,987 sentences) has better accuracy than BERT fine-tuned on a smaller hand-labelled dataset *with* HTML (our Persons evaluation set, 1,214 sentences) included. For this reason we use BERT for Intek, as to improve our approach would require the in-

vestment of considerable resources in labelling over ten times more data.

In the next chapter we move on to the evaluation of Intek in our interviewing study.

5

INTEK STUDY, RESULTS AND DISCUSSION

This chapter presents results from our empirical job interviewing study. We first introduce the study aims, participants, conditions and overall process. We then describe the data collected from the study and present our high-level results. We then discuss our other results in an order that demonstrates Intek's contribution to deception detection. Firstly, we show Intek was capable of returning useful information for all interviewees in the hands of an expert. Secondly, we compare real interviewer Intek usage compared to expert usage. Thirdly, we show how this usage corresponded to the essential interviewer skills required for deception detection: control of the interview; test questioning; judgement of deceptive behaviour. We also present a usability analysis of Intek including reported issues and suggested solutions.

5.1 INTEK STUDY

The overall aim of this study is to determine if Intek can improve deception detection performance in job interviewing over the two baseline approaches "standard" and CCE.

The study was designed to collect data from the study results, various questionnaires, Intek application logs, expert ratings, video and audio interview transcripts. This data enables us to "drill-down" into the overall results and examine them in detail to provide a picture of Intek's end-to-end performance, from data source to question delivery.

It should be noted that this chapter focusses on Intek and the reasons behind its performance. We do not specifically discuss the results of the standard or CCE conditions; they are included as comparative baselines for Intek only. Performance of standard and CCE are explored more fully in the upcoming thesis Sweeney (2022).

We now supply more detail on the study participants and conditions before going on to describe the study process.

5.1.1 *Participants*

This study conducted one-on-one job interviews using a single interviewer and interviewee. Thirteen novice interviewers interviewed 111 interviewees over the three conditions. A small admin team was required to process participants, highlight CV items for discussion in the interviews and to manufacture and control the quality of lies used by deceivers. The only incentive given to interviewers and interviewees was the interviewing/interview experience gained and CV feedback for interviewees given by the admin team. We now give some more detail on these participant groups.

5.1.1.1 *Interviewees*

Each interviewer accepted the aim of recruiting four interviewees per condition. These interviewees could be recruited from any source, mostly friends, family and colleagues, although indiscriminate social media campaigns were used by some interviewers. This total was supplemented by "reserve" interviewees introduced by the admin team to make up the total. Interviewees for each condition were randomly allocated to an interviewer and a truth-teller or deceiver group. The final interviewee counts for each condition by truth-teller/deceiver are shown in Figure 5.1. Several interviewees dropped out after allocation, also five interviews were reallocated from the standard condition to the CCE condition as the interviewers had demonstrated CCE-like techniques during these interviews. These factors resulted in the higher interviewee total for the CCE condition.

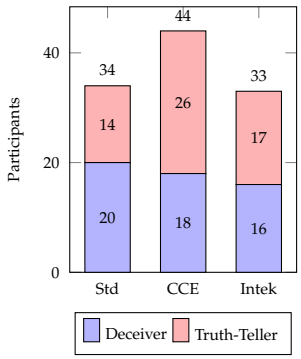

Figure 5.1 Final interviewee counts by role and condition

We aimed for a similar distribution of interviewee experience and occupation across conditions and particularly within the deceiver condition. The deceiver condition is more difficult for an interviewer

to correctly identify than truth-teller due to truth bias among other factors. Figures 5.2 and 5.3 show these distributions broken down by truth-tellers and deceivers. These breakdowns show an unavoidable increase in higher experience for Intek truth-tellers, as by this point some of our more experienced reserve interviewees were being called on. However, the more important deceiver group is quite well balanced.

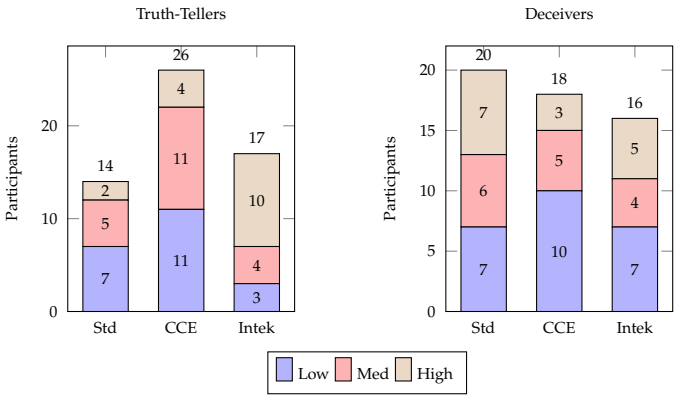

Figure 5.2 Final interviewee counts by condition and experience level split by role

Figure 5.3 Final interviewee counts by condition and employment type split by role

5.1.1.2 *Interviewers*

Interviewers were recruited from psychology undergraduate and postgraduate students, all gaining potentially useful interviewing experience through this study. Thirteen interviewers participated in total. Eleven of these interviewers started at condition one; two more joined at condition two. All thirteen can be considered novice interviewers, although two had previously been trained in interviewing but had

not carried out any interviews. The aim was for each interviewer to recruit twelve interviewees each (or eight for the later starters).

Recruitment of interviewees proved difficult for some interviewers. The main problems quoted by those who declined to interview were lack of time for the interview tasks, lack of interest in or intimidation by the task itself. The only incentive offered to interviewees was job interview experience. It is possible financial incentive may have attracted more interviewees and we will trial this approach in any future studies.

Interviewers actually recruited between two and thirteen interviewees (on average recruiting 8.7), we therefore supplemented the study with twenty interviewees from an admin team pot; these participants are included in Figure 5.1, 5.2 and 5.3.

5.1.1.3 *Admin Team*

The Admin Team consisted of two researchers, including the author, and one psychology professor. Our main responsibilities included: assuring the standard formatting of CVs; highlighting CV items for discussion; creating three lies per CV for deceivers and integrating them seamlessly back into the CV; and the not inconsiderable task of general administration of all participants throughout the study lifecycle (see section 5.1.3). The lie manufacturing was particularly sensitive and required the approval of all team members.

5.1.2 *Conditions*

We now describe the three experimental conditions that made up the study: standard; CCE; Intek. The key elements of these conditions are compared in table 5.1.

Condition	Interview Structure	Active Listening	questioning Technique	Test Variety	Probing Tests	Unexpected Tests	Veracity Testing	Behaviour Observation	Research
Std	WASP	Yes	STAR	Narrow/scripted	No	Scripted	None	None	Manual
CCE	CCE	Yes	CCE	Unlimited/manual	Yes	Er generated	In principle	Against baseline	Manual
Intek	CCE (via Intek)	Yes	CCE (via Intek)	Wide/automatic	Yes	Intek generated	Direct	Against baseline	Automated

Table 5.1: Comparative features of the three study conditions

5.1.2.1 *Condition 1: "Standard"*

The standard condition is based on interview methods used by the UK chartered institute of personnel and development (CIPD) and is designed to emulate a normal job interview that might take place

in industry. The scenario we use is that interviewees are interviewing for a job similar to their current role. Interviewers received an initial training session in which they learnt the theory and practiced the techniques mentioned below. They also received a "top-up" training session approximately half-way through the condition. We now provide information on the main three techniques used in this condition: active listening; WASP; STAR.

The **Active Listening** technique brings structure to interviewer listening and response throughout all stages of the interview. Active listening asks interviewers to be aware of, understand and try to mitigate distrations such as: their own daydreams poor focus or assumptions and prejudices; interviewee cultural, language or other characteristics that interviewers might find annoying or biasing; environmental factors such as comfort, noise or visual distrations. The key points of active listening for interviewers are as follows:

- Pay attention: look at interviewee directly, put aside the various types of distractions, listen to the interviewee verbally and observe body language.
- Show that you are listening: use body language and gestures to convey attention, using an open and inviting posture. Encourage the interviewee with small verbal comments and phrases.
- Provide feedback: reflect on what the interviewee has said by periodically paraphrasing and asking questions for clarification.
- Defer judgement: do not interrupt the interviewee with judgements or counter arguments.
- Do not intimidate: nothing is gained by intimidating the interviewee. Active learning aims to be a model for respect and understanding, while trying to gain information and perspective. Be candid and open, try and treat the interviewee as you would want to be treated.

WASP brings structure to the overall process of the interview, in theory allowing the interviewer to gain the most information out of interviewee in the short time available. The prescribed phases are as follows:

- **W**elcome candidates: the interviewee should be able to relax and make contact with interviewer, fostering a relationship of trust and comfort.

- **A**cquire information: use the STAR model for questioning.
- **S**upply information: give the interviewee the chance to clarify or seek more information. Interviewee questions give insight into their preparedness.
- **P**art from candidates: end on a friendly note, thank the interviewee, aim to give the interviewee a positive memory of the interview.

STAR brings structure specifically to the questioning/information acquisition stages of the interview. This is done by providing types of questions in a usable sequence to ensure interviewers cover all relevant information. STAR also advises interviewers to use test questions based on interviewee responses. Interviewers are asked to control talkative interviewees by bringing them back to the desired topic. The prescribed question types are as follows:

- **S**ituation: what situation was faced by the interviewee?
- **T**ask: what part did the interviewee play?
- **A**ction: what specifically did the interviewee do?
- **R**esult: what was the outcome? Was the outcome as intended?

In practice, the standard condition is not expected to perform much above chance for deception detection. The variety of tests are quite narrow, with the same tests being used in the same order for each topic (job role). There is no prescribed veracity checking and probing tests are limited. Standard techniques are not specifically aimed at deception detection, relying on intuition to spot deceivers, rather than baselining and behaviour change as in CCE conditions. Interviewers were encouraged to perform around twenty minutes of their own research into the candidate's CV. For this condition, research must be undertaken, collated and accessed during the interview manually in any way the interviewer sees fit.

As previously mentioned, five interviews from the standard condition were reallocated to the CCE condition due to CCE-style test questioning being used. We conjecture this is because certain interviewers had studied the CCE literature before the CCE condition commenced. The effect of this re-allocation was to increase CCE results slightly, however the Intek condition is not effected at all.

5.1.2.2 *Condition 2: CCE*

The controlled cognitive engagement (CCE) (Ormerod and Dando, 2015) condition aims to provide a natural and friendly interview using rapport building and active listening techniques, but adds a framework of iterative information gathering and probing test questioning. This enhanced questioning is designed specifically to increase cognitive load in deceivers to enable deception detection. CCE deception detection techniques are derived from investigative interviewing concepts such as the PEACE model, the cognitive interview and theories of memory. CCE provides a framework which makes active use of the interviewer to generate probing test questions. These questions aim to make the interviewee work harder, both talking and thinking more than the interviewer. The more the interviewee works, the more the interviewer has opportunity to actively listen and observe the interviewee for deviations from the norm, which may be evidence of deception. The CCE framework includes past, present and future aspects to questioning and a randomisation of topics for unpredictability. CCE is discussed in more detail in chapter 3 section 3.1.

The interview scenario is the same as the standard condition; that interviewees are interviewing for a role similar to their current role. Again, interviewees received an initial training session in which they practiced the techniques mentioned in this section, as well as a "top-up" session approximately half-way through the condition.

The phases of a CCE interview from an interviewer perspective follow the **BITE** protocol as follows:

- **B**aseline: open a dialogue, build rapport, observe normal verbal and physical behaviour with neutral questioning and active listening.
- **I**nformation gathering: gather information from the interviewee, control the interview, commit the interviewee to a version of the truth. Use tell, explain, describe (TED) open questioning with active listening, varying topics.
- **T**est the account: make use of the information gathered to test for veracity, allowing the interviewer to reach a confidence decision. Tests should be veracity testable *in theory*, not in practice, as the interviewer is unlikely to be knowledgable in the same areas as the interviewee. Use who, what, where, when, why, how (5WH) questions, aiming for *unexpected tests of expec-*

ted knowledge. Again, use active listening to encourage further, deeper information and observe for behaviour change.

- **E**valuate: the interviewer should be able to reach a confidence decision on a topic after two or three tests.

We expect the CCE condition to perform better than chance, as it is specifically aimed at deception detection. CCE packages up deception detection in an easy-to-remember framework and employs probing test questioning using a good variety of unexpected and, in theory, veracity checkable questions. However, a lot is being asked of novice interviewers: they must generate unexpected tests of expected knowledge, while simultaneously keeping track of the interview and observing for behaviour change. Also, CCE uses observation for behaviour change as the sole method of deception detection. This is understandable, as it is likely interviewers will not be knowledgeable enough about most items of discussion to veracity check answers. This is consistent with a general recruitment agency scenario.

As with the standard condition, interviewers were encouraged to undertake twenty minutes of research per interview, which was then collated and used during the interview manually.

5.1.2.3 *Condition 3: Intek*

The Intek condition uses the same interviewing techniques as CCE: baselining; iterative topic-based information gathering; probing test questioning; use of active listening and observation for behaviour change. The Intek condition is in fact CCE *plus* Intek. Intek adds a centralised interview structure displayed on-screen, rather than on paper or held in an interviewer's memory. Information gathering questions and related probing test questions are generated for interviewers and displayed in-topic on-screen. Test question information is presented as facts, thereby questions based on these facts are directly veracity testable. This veracity testing adds another route for deception detection along with the observation of behaviour change. Intek theoretically has less variety of tests than CCE, as CCE tests are generated by interviewers and so are technically unlimited, but Intek provides a wide selection of test information covering different aspects of a topic and different types of information. Intek "researches" topics automatically, once an information snippet is input. Selecting which questions to ask can be onerous, but the process does serve

to familiarise the interviewer with the subject matter in a structured manner.

Interviewers received an initial training session in which technology pre-requisites and setup were covered, along with a hands-on practice of CCE using Intek on real CVs in a realistic interview scenario. Interviewers used their own computers with Google Chrome to run Intek, with Zoom windowed over the top-right portion of the browser to interact with the interviewee. We recommended that no CV was used during the interview, with Intek relied upon to guide the interview. No mid-condition "top-up" training session was required for Intek. Minor fixes were performed on Intek throughout the condition, no major functionality or sources of information were added.

Using Intek, interviewers should find generating test questions and keeping track of the interview overall much less cognitively demanding. Both of these benefits should leave more time available to observe for behaviour change and selecting ideal test questions to maximise cognitive load for deceivers. Good tracking of interview progress should also enable interviewers to better control the interview, ensuring interviewee's answer the desired test questions. Intek veracity testing may catch out deceivers directly. For these reasons, we expect Intek to perform better than CCE alone in the majority of interviews, especially where good data is available to cover input snippets. However, multiple potential difficulties remain:

- Intek may not be able to generate a useful level of questions for some searches. Information may be either too vague, too obscure or missing completely.
- Intek may not be usable if it encounters technical failures or a failure of the Zoom/Intek concept as a whole prevents adoption or use as planned.
- The presence of Intek may cause interviewer complacency and thereby lack of effort; effort is still required with Intek to pick good questions, as asking questions in order, without thought, can sometimes cause inappropriate questions to be asked and the interviewer to then lose credibility and control.

We find in the results in Section 5.2 that Intek did not suffer any great problems, achieving good deception detection results. We now move on to discuss the study process.

5.1.3 *Process*

The entire study process, from the perspective of the Admin Team, including the recruitment of interviewers and interviewees, to the end of the study is shown in figure 5.4 and described below.

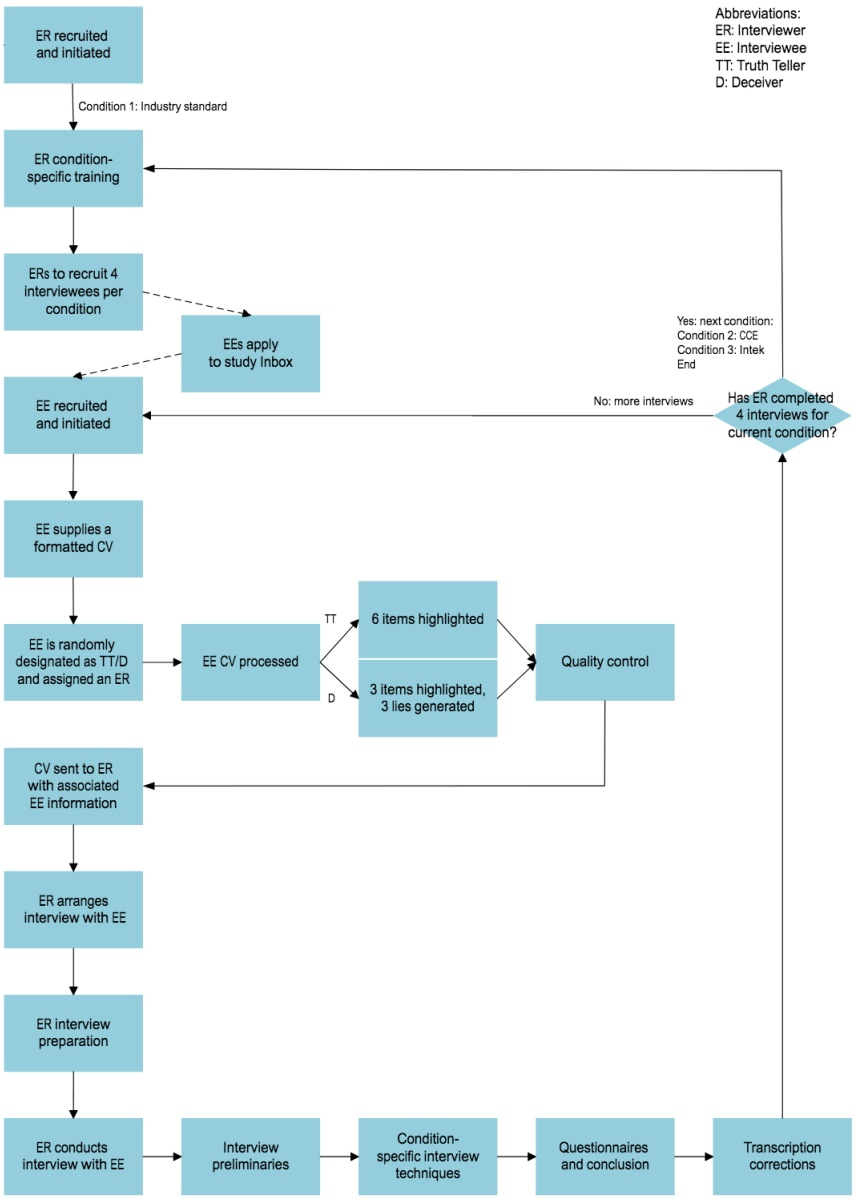

Figure 5.4 The end-to-end process for the whole study, from the perspective of the Admin Team

Interviewers were intially recruited using an advert designed to explain the study premise, the burdens and benefits of participation. If they wished to continue, they were given futher information then signed an online consent to participate, at which point a "kick off" ses-

sion was held for all interviewers to recap and introduce participants to each other.

The standard condition then began with a training session for the initial eleven interviewers. Interviewers were introduced to the condition as described in Section 5.1.2 and were shown the documentation lifecycle for themselves and interviewees. Documentation for interviewers consisted of: a detailed guide for every step in their process, including technology setup; interviewee recruitment and their documentation; how to use Zoom; a recap of the condition concepts from the training and how to apply them in an interview; which questionnaires to use after the interview; how to correct the audio auto-transcription from the interview. This guide was updated with each condition to include the relevant techniques, including a guide for Intek use for the Intek condition.

Interviewers then started to recruit their first four interviewees using another advert which gave information and prompted potential participants to contact the study inbox. The inbox was monitored by the admin team and any participants responded to with information and consent forms along with CV instructions. CVs had to include a home postcode, employment roles, education and interests on two sides of A4 (maximum) in 11pt font size (minimum). This standard CV format aimed to remove potential bias by the use of irregular elements or extra detail. Once a satisfactory CV had been received and the interviewee consented to the study, they were assigned an anonymous identifier which would then be used for the remainder of the process and their CV was anonymised using this identifier. Interviewers were anonymised in a similar fashion.

Interviewees were then randomly assigned as a truth-teller or a deceiver and their CV processed accordingly. For a truth-teller the CV process was straightforward: six items from their CV were highlighted for discussion in the interview. These items might be a home postcode, a job role, a company or some other aspect of that job, an aspect of their education or an interest. These six items must be discussed in the interview, and the interviewer's final deception decision for the interview was based on those items. The topic of home was included as something candidates anecdotally lie about, that can have a large impact on an employee's ability to consistently attend a work location and perform their duties. Interest was included as a topic as this is typically discussed in interviews in relation to a candidates "fit" for an organisation or team. Job role and education related items

are included for obvious reasons. For deceivers, the process was more complex in that three items were highlighted, plus three lies manufactured and inserted into the CV. Interviewers were told any number, but at most three, of these highlighted items might be lies, in order to prevent them from guessing the method. Clearly this made the task more difficult for interviewers but this is in line with a real-world job interviewing scenario. The three lies may cover any of the topics mentioned previously.

Clearly the outcome of the study is sensitive to the consistency of highlighted items and the quality of lies introduced. For this reason truth-teller CVs were approved for consistency by one other member of the Admin Team and deceiver CVs approved by two members to ensure the lies properly fit the narrative, plus being sent for feasibility and approval to the interviewee. Lie manufacturing was subject to guidelines which formulated how much of a stretch each lie should be for an interviewee. The interviewee was encouraged to undertake whatever research they deemed necessary to carry off the lie as truth.

Figure 5.5 shows the distribution of topics chosen for highlighted CV items, both truths and lies, across the three conditions. Figure 5.5 uses the six Intek topic types to categorise the types of CV items highlighted; of course natural CV items do not all fit cleanly into these six types, so we have allocated highlighted items to the closest type. The distribution of CV items we select is somewhat random in that these items should fit with a natural interview scenario; the six most significant, important or interesting points are selected across a reasonable range of topics for discussion. In general a single home topic is selected if other significant items are lacking, then approximately four organisation or course topics are selected depending on the distribution of work experience or education in the CV, then finally one or two interests to make up the six. Figure 5.5 shows the distribution of highlighted CV item types are fairly uniform across conditions.

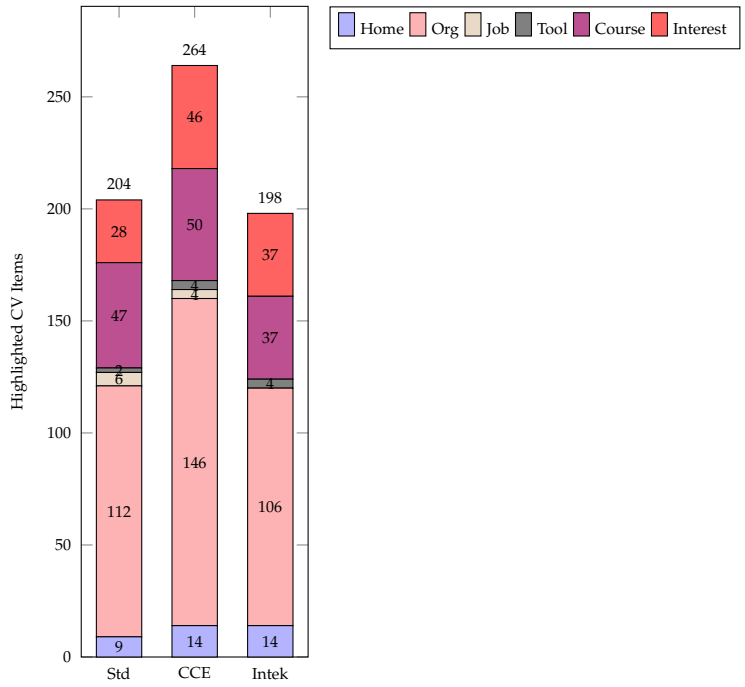

Figure 5.5 Highlighted CV item counts by condition and the Intek topic used to search for that item.

Once the CV had been processed and approved, it was randomly assigned to an interviewer, with other essential information such as interviewee contact details, to arrange, prepare for and carry out the interview. Interviewers were instructed to reject any interviewees they had prior knowledge of. The interviewer conducted the interview using the relevant techniques from the current condition, with reference to the provided detailed guide.

After the interview the interviewer and interviewee both completed separate quesionnaires which included the interviewer's deception judgements overall and on the six highlighted items in the CV individually. Interviewers then corrected audio transcriptions before repeating this whole process for, in theory, four interviewees per condition and three conditions in total. In practice, some interviewers were not able to supply four participants per condition or complete all their interviews before each condition cut off, in which case they moved any remaining interviews to the next condition. Two additional interviewers joined the process for the CCE condition.

Finally to complete the study, interviewees and interviewers were sent a debrief document and interviewers were asked to complete a post-study questionnaire which included Intek feedback.

We have described our job interviewing study in detail and now move on to presenting results with analysis.

5.2 INTEK RESULTS AND DISCUSSION

In this section we list the data we have gathered from which we extract our results. We then discuss high-level results and then other results in an order that demonstrates Intek's contribution to deception detection. Firstly, we show Intek was capable of returning useful information for all interviewees in the hands of an expert. Secondly, we compare real interviewer Intek usage compared to expert usage. Thirdly, we show how this usage corresponded to the essential interviewer skills required for deception detection: control of the interview; test questioning; judgement of deceptive behaviour. We also present a usability analysis of Intek including reported issues and suggested solutions.

5.2.1 *Data Sources*

All results and analysis are sourced from the following materials.

5.2.1.1 *Interview quality review*

An expert interviewer reviewed all interview video recordings, scoring the interviewer for interview quality in six areas: all highlighted CV items covered; rapport building and professionalism; active listening and feedback; appropriate questioning; flow and transitions; adherenece to method. Qualitative feedback was also given for interviewer and interviewee behaviours.

5.2.1.2 *Intek usage quality review*

An expert Intek user repeated the interview preparation for all Intek interviews using the same CVs as interviewers received. This gives us a comparable Intek performance baseline for the quantity of searches required and the quantity of suitable factoids returned for an optimal interview.

5.2.1.3 *Interviewee questionnaires*

This questionnaire was completed by interviewees straight after the interview. It records a few personal attributes, four questions regarding perceived interview difficulty and self-rating their own performance, using a seven-score Likert scale (Likert, 1932) to capture direction and strength of opinion.

5.2.1.4 *Interviewer post-interview questionnaires*

This questionnaire was completed by interviewers straight after an interview, but before correcting the interview transcript to avoid any infuence this may have had on their deception judgement. It contains fifteen questions which are a mix of seven-score Likert scales, qualitative text and other selections. The Intek condition added an additional ten questions regarding the usability of the Intek application. Note that the ten Intek questions were missed by one interviewer, so these results total 32 rather than 33.

5.2.1.5 *Interviewer post-study questionnaires*

This questionnaire was completed by interviewers after the whole study was completed, mainly to collect information about Intek. It contains twelve questions regarding aspects of Intek, using seven-score Likert scales and qualitative text. It also contains a standard ten-question usability evaluation using the system usability scale (SUS) (Brooke et al., 1996). Only eight of the thirteen interviewers completed this questionnaire, therefore the results may be somewhat biased toward this groups' motivations.

5.2.1.6 *Intek logs*

Intek was created with future analysis in mind, so most user actions were logged. We compare the logs with expected activity from the expert Intek reviews above to assess the quality of Intek usage.

5.2.1.7 *Interview transcript analysis*

Automatic transcriptions for all interviews were corrected by interviewers. Analysis of interviewee to interviewer word count ratios, question counts and word counts are used.

We now move on to discussing the results generated from these data.

5.2.2 *High-Level Results*

	Interviews conducted			Highlighted CV items		
Condition	TT	D	Total	TT items	D items	Total
Std	14	20	34	144	60	204
CCE	26	18	44	210	54	264
Intek	17	16	33	150	48	198

Table 5.2: Study conditions, interviews and highlighted CV items count by truth-teller (TT) and deceiver (D). Each interview had a total of six CV items highlighted, for deceivers three of these items were lies.

Table 5.2 shows the final number of interviews conducted under each condition and the number and type of associated highlighted CV items discussed in those interviews. As previously outlined, these numbers are somewhat unbalanced due to interviewee drop-outs, recruitment difficulties and the carry-over of five interviews from standard to CCE. However, we have tried to balance the deceiver numbers as much as possible as these are key for detecting deception. Note there are many more truth-teller (TT) highlighted CV items than deceiver (D) items, as TT interviews consist of six TT items and D interviews three TT and three D items.

Condition		Actual TT	Actual D	Accuracy
Std	Predicted TT	9 (64.3%)	13 (65.0%)	16 (47.1%)
	Predicted D	5 (35.7%)	7 (35.0%)	
CCE	Predicted TT	19 (73.1%)	12 (66.7%)	25 (56.8%)
	Predicted D	7 (26.9%)	6 (33.3%)	
Intek	Predicted TT	**13 (76.5%)**	**5 (31.3%)**	**24 (72.7%)**
	Predicted D	4 **(23.5%)**	**11 (68.8%)**	

Table 5.3: Study results for the interviewer *overall* deception judgement of truth-teller (TT) or deceiver (D).

Table 5.3 shows results for interviewer judgement of deception for the interview overall. These results were gathered from interviewer post-interview questionnaires.

Intek shows a significant increase in performance for all measures; overall accuracy and thereby true positives for truth-tellers and deceivers are increased, as well as a corresponding decrease in false positives. These results show an increase in accuracy with each condition, however, the most striking result is the increase in deception de-

tection for Intek, around twice that of either baseline condition. This result is impressive due to the difficulty of the deception detection task. Interviewers are more likely to make a truth decision due to a number of factors: truth bias; the task itself is based on interviewees trying to persuade interviewers they are telling the truth; also interviewers see far less lies and deceptive behaviour so are less calibrated to it, this is especially true for novice interviewers. All these factors are difficulties in real-life interviewing. This non-linear two-fold improvement in deception detection for Intek is a strong indication that this improvement is due to the introduction of the Intek technology, rather than a gradual improvement in interviewer skills.

The accuracy of the standard and CCE conditions are slighly below and slightly above chance respectively, which is in-line with the performance of traditional methods in literature (Bond Jr and DePaulo (2006)). This slightly disappointing performance for CCE is not analysed further as it is outside the scope of this thesis, however CCE performance is examined further in Sweeney (2022). We mention accuracy here to give some comparison to literature, but it should be mentioned that as all our conditions contain a different inbalanced ratio between truth-tellers and deceivers and that as predicting truth-tellers is easier than predicting deceivers for the reasons discussed above, accuracy should only be used as a guide. For instance, CCE, with the lowest deception detection performance of 33.3% has an inflated accuracy of 56.8% due to its high ratio of truth-tellers. However, this inbalance is not severe.

Condition		Actual TT	Actual D	Accuracy
Std	Predicted TT	128 (88.9%)	52 (86.7%)	136 (66.7%)
	Predicted D	16 (11.1%)	8 (13.3%)	
CCE	Predicted TT	188 (89.5%)	45 (83.3%)	197 (74.6%)
	Predicted D	22 (10.5%)	9 (16.7%)	
Intek	Predicted TT	**144 (96.0%)**	**25 (52.1%)**	**167 (84.3%)**
	Predicted D	**6 (4.0%)**	**23 (47.9%)**	

Table 5.4: Study results for the interviewer deception judgement for *individual highlighted CV items*.

Table 5.4 shows results for interviewer judgement of deception for the individual highlighted CV items. These results were gathered from interviewer post-interview questionnaires.

Spotting deception for individual items is a difficult task for the same reasons as spotting overall deception. For the overall judgement,

the interviewer has three highlighted lies for a deceiver, so clearly they get three attempts per interview to spot deception (although the interviewer is not aware of this). For individual items, the interviewer gets only one attempt, hence the lower scores for deception detection in table 5.4. This difficulty increases the bias toward truth-teller prediction even further, with high correct TT predictions for all conditions, but especially Intek which scores highest. The standard and CCE conditions pay for this increase in TT prediction with low deception detection results, with many more incorrect deceptive than correct judgements. The Intek condition reverses this deception detection trend, with low deception false positives and a three-fold increase over the baseline conditions for correct deception judgements. Increases of correct TT prediction and overall accuracy for Intek over baseline conditions are less dramatic but still substantial. As with the overall results, the increase in correct deception judgements and decrease in deception false positives using Intek is so dramatic that it strongly indicates the Intek technology as cause. The only substantial change for the Intek condition over CCE is the Intek technology itself.

We now move on to analyse the key factors underlying these results focussing on four areas: the performance of Intek when used as intended; a comparison of intended use with actual use in our study; an analysis of Intek's contribution to the underlying tasks in deception detection; an evaluation of Intek usability.

5.2.3 *Best-Case Performance*

The first step in assessing how well Intek supports interviewers in detecting deception in job interviewing, is to analyse how well Intek is *capable* of performing in the "best-case" when used as intended. This analysis gives us a chance to recognise any particular deficiencies in Intek's functionality and also gives us a benchmark for comparison against actual use by interviewers in our study in the next section.

We assess Intek performance when used as intended by performing a full interview preparation using the same CVs as used in the Intek condition in the main study. These preparations are performed by an Intek expert user (the author) aiming for the same preparation times as interviewers (thirty minutes to an hour).

In this section we are assessing the searches performed and the information returned by Intek during interview preparation. We exam-

ine how this information is used by interviewers during the interview in section 5.2.5.

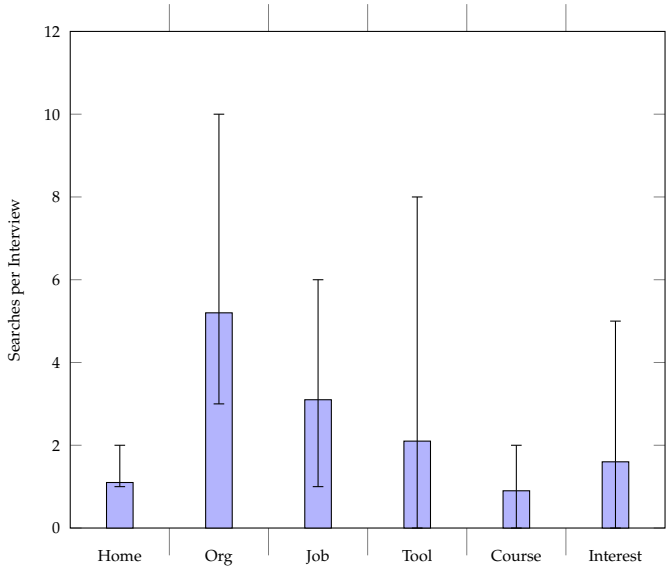

Figure 5.6 The mean, minimum and maximum number of Intek searches performed of each topic type in a single interview (CV).

Figure 5.6 shows the mean, minimum and maximum searches for each topic type, performed by the expert Intek user for all interviews in the Intek condition. Each interview has exactly six highlighted CV items. As previously shown in Figure 5.5, the majority of highlighted items are Organisation topics which generally refers to a job role and thereby will probably benefit from associated Job and Tool topic searches. The next most popular topics searched are Course and Interest, followed by Home.

The mean values shown in Figure 5.6 are generally as expected, as they are close to the proportions of topics from corresponding CV items mentioned above. This means that in general Intek searches are functioning as designed. The Home topic values are exactly as expected. The Organisation topic minimum and mean are exactly as expected, although the maximum of ten indicates some CV Organisations require multiple searches to locate the correct entity, for example "Costa" returns too general a result, requiring an extra search for "Costa Brighton". The fact that the Organisation mean is relatively low indicates most searches are correct first time. The Job topic is expected to be slightly lower than Organisation for mean, minimum and maximum as an Organisation search also preceedes a Course search. The Job minimum is lower than expected as we find some job roles repeat within a CV and some roles are very generic, for example

"student", which reduces the searches performed. Course and Interest topics are as generally as expected; both topics have zero minimums as these topics do not appear on every CV.

We now move on to look at the information factoids returned from these searches.

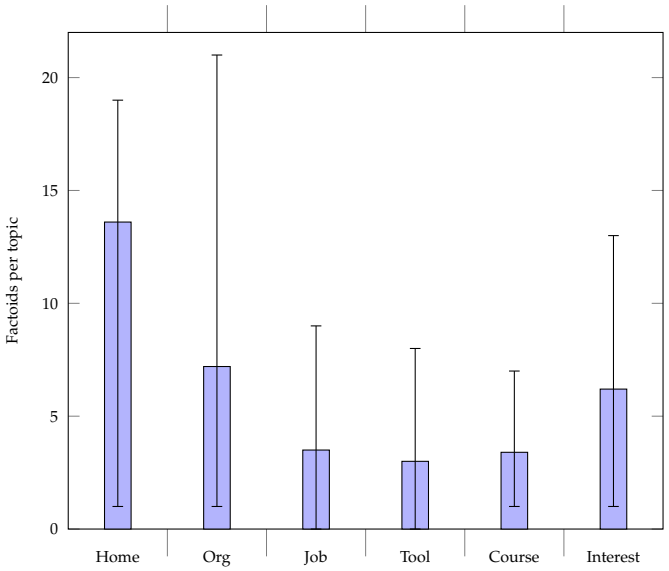

Figure 5.7 The mean, minimum and maximum of number of useful factoids delivered by Intek searches for each topic type in a single interview.

Figure 5.7 shows the mean, minimum and maximum number of useful factoids returned in response to a single Intek search, for each topic type, performed by the expert Intek user over all interviews in the Intek condition. Only factoids that appear directly useful in test questioning in the context of the current CV item are counted.

This is an important graph as it shows whether Intek actually returns useful information over real-world CVs.

Overall, the mean values are as expected. The Home topic, due to its geographic nature, contains a relatively high number of useful factoids. The Organisation and Interest topics use a variety of different sources and so contain a good number of factoids. Job and Tool topics rely heavily on frequently-asked question and answer factoids and so return the least number of useful factoids. However, this is not a problem as Job and Tool are designed to be used in tandem with an Organisation search. For reference, CCE indicates that two or three, to a maximum of five, good test questions should be sufficient for a CV item.

The maximum values give an indication of the full potential of a topic for test questioning. Examples of searches that return the maximum number of factoids for each topic are as follows. For the Home topic, a postcode that represents a medium-sized built up area, for example St. Albans, as this contains points of interest of all types and is likely to have a Wikipedia summary in place. For the Organisation topic, an entity such as the University of Sussex has a well populated Wikipedia "infobox" containing several useful facts, as well as good geographical and other information. For Job, a role, such as "occupational therapist assistant", that is popular enough to be frequently advertised for, but not so generic that the information returned is obvious general knowledge is ideal. The Tool topic is conceptually similar to Job, in which a search such as "brochure production cycle" returns the maximum factoids. For the Course topic, an institution with a well formatted list of modules as well as a web site mentioning key people or a specific location in which study takes place, performed well. The Interest topic is similar to Job and Role, in that a middle obscurity search, such as "kite surfing", is ideal, but includes several extra data sources, such as nearby clubs, that increase the total of useful factoids.

The minimum values are zero or one across all topics. As the means are relatively high, indicating good performance in general, these low minimums are acceptable, as we do not expect to be able to deliver great information for every single input snippet. Few factoids are returned when topics are used in an exploratory or hopeful fashion, not quite as designed, for example an Interest such as "Thomas International Psychometric Profiling" or a location such as "Saint Petersburg, Russia". More frequently, an obscure entity does not provide a big enough target for information retrieval, resulting in few factoids returned, for example the Interest "Video Juke Boxes". Conversely, a search may be so general that factoids returned are obvious general knowledge, for example the Job role "Assistant".

The results discussed so far describe Intek searches and factoids returned by individual topic. In order to examine the quality of whole Intek interviews, which are a combination of the number of searches performed and the number of usable items returned, we introduce a scoring system for each interview. This system rewards multiple searches returning at least three usable factoids. The score awards one third for each usable factoid returned for each topic, up to a maximum score of one for each topic. A whole interview should contain

six topics as an absolute minimum. As an example, six topics returning at least three items would give a total score of six. Thirds are used as three is a minimum number of factoids required for basic CCE test questioning; using thirds also limits the compensation of one topic with few searches by others with many searches. These scores should give a balanced view of the whole interview based on the minimum requirements.

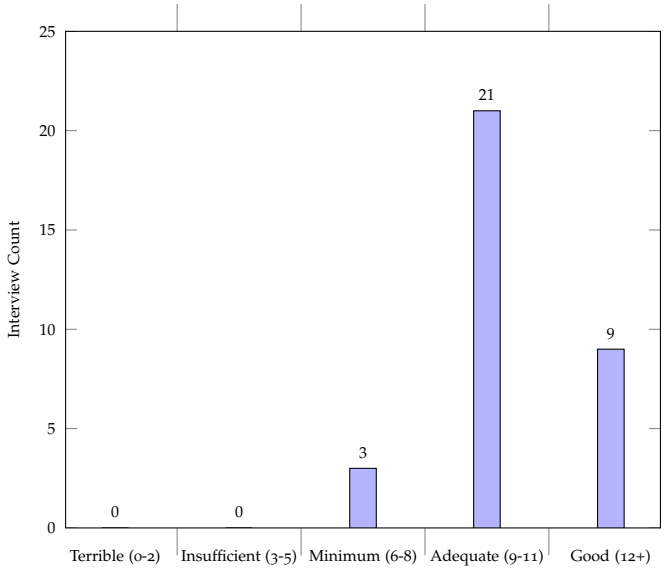

Figure 5.8 Counts of interviews by interview quality. Quality is scored by combining the number of Intek searches with the number of usable factoids returned. Each interview quality group is shown with its score range.

Figure 5.8 shows the number of Intek interviews that fall into each interview quality score category. These results show that, when used by an expert, Intek performs well; apparently returning enough usable factoids for a good range of topics to perform at least an adequate CCE interview in 91% of interviews.

The three interviews in the Minimum group each score 8.67, so are only just short of an adequate score. Investigation into these three show a number of issues that causes the relatively poor scores: short sparse CVs containing few details for extra searches and CVs containing generic roles which are difficult to extract specific enough information, for example Barista, Volunteer, Waiter. Both these issues are difficult to deal with and are discussed further with potential solutions in Section 5.2.6.

Having ascertained that Intek has performed well and as designed overall, we now move on to explore how well it was used by interviewers during the study.

5.2.4 *Best-Case Comparison with Actual Use*

In this section we use the search, factoids returned and quality score results from the previous section as an "expected" score for comparison against how users performed in the actual study interviews. We also examine users' views on these aspects of Intek extracted from the interviewer questionnaires.

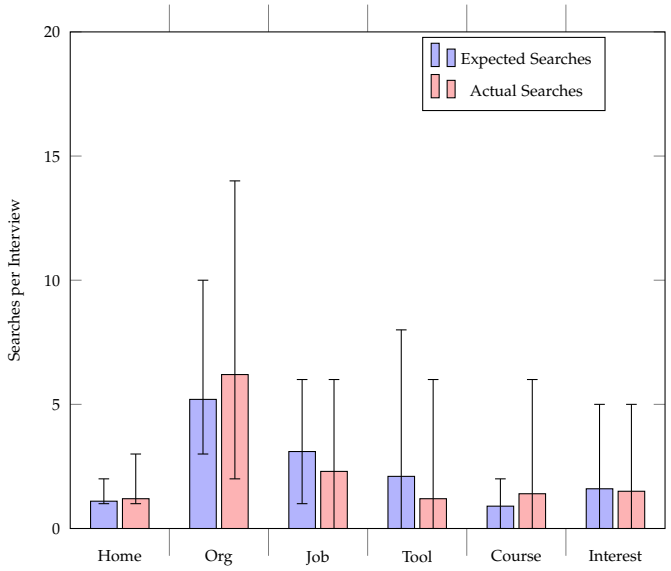

Figure 5.9 The mean, minimum and maximum number of Intek searches performed of each topic type in a single interview (CV). Expected values from the previous section are compared to the actual searches performed by users in genuine Intek interviews extracted from Intek logs.

Figure 5.9 compares the expert users' expected Intek searches from the previous section against our novice interviewers' actual searches during the Intek condition of the study extracted from Intek logs. This allows us to examine whether Intek is being used effectively.

Overall the means are very close between expected and actual, which indicates no cause for concern and that users are following their training. In general using a good breadth of topics to cover questioning for all highlighted CV items. However, some deviances from expected may be observed. The Organisation topic mean and maximum are higher than expected, which can be traced back to

two interviews in which the interviewer struggled to find a good match, in particular one company based in California, in which the interviewer tried various suffixes: CA, california, Inc etc. The fact that overall means are close to expected is a good indication that all topic searches are generally finding the desired entity or concept with the first search. The Organisation minimum is lower than expected due to a single, apparently rushed, interview in which the interviewer missed two Organisation searches for unknown reasons. Job and Tool topics have slightly lower than expected means and minimums, indicating interviewers did not make quite as much use of the cross-referencing potential these topics as they could have. The Course topic has a high maximum due to a single interview in which the same search was inexplicably repeated five times.

We now examine how users dealt with the factoids returned from these searches.

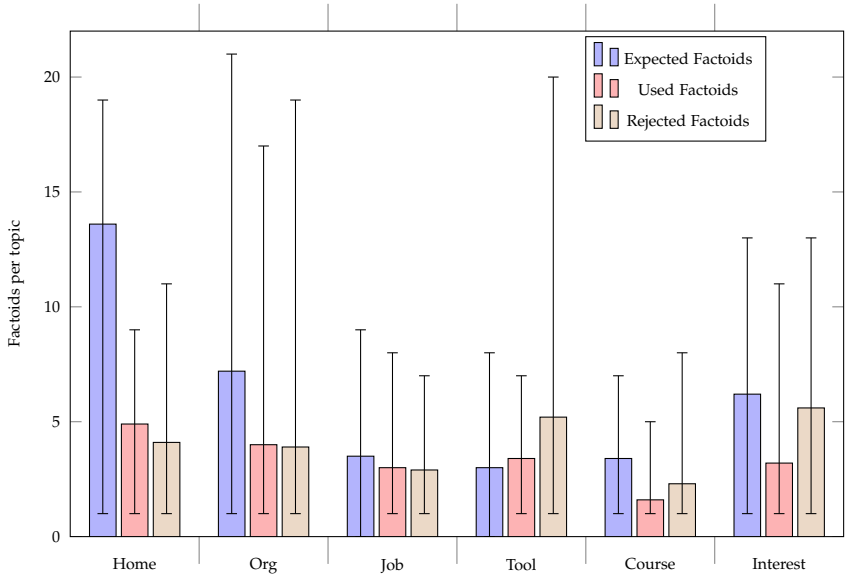

Figure 5.10 The mean, minimum and maximum of number of useful factoids delivered by Intek searches for each topic type in a single interview. Expected values from the previous section are compared to factoids apparently used for questioning or rejected for questioning by interviewers.

Figure 5.10 compares the expert users' expected Intek useful factoids from the previous section against our novice interviewers' used and rejected factoids from their preparation for interviews during the Intek condition of the study, extracted from Intek logs. This allows us to examine the quality of information returned by Intek and how this information is used.

We ascertain used and rejected factoids by analysing Intek logs. A used factoid is indicated by being opened and left open by a user, whereas rejected factoids are subsequently closed by the user. We filter out any factoid closing that is done after the interview is complete.

Figure 5.10 shows that overall, used factoids means are in the three to five range for all topics, with the exception of Course. Course has one particularly detailed factoid, which is typically used in isolation without checking other factoids, so the Course mean is expected to be lower. These used factoids means of around, or just below, five factoids with correspondingly high or higher rejected factoids indicates users are making a good effort to review factoids and are stopping when they have found around five factoids that they deem suitable for test questioning. Interviewers have been told in training that five test questions should be sufficient for CCE; they appear to be taking five as a stopping point for factoid review. It should also be taken into account that factoids may contain more than one suitable test question, but this is entirely variable and depends on the search used and the factoid type. For example, a Wikipedia infobox might contain five or more useful facts. In general these means show that Intek is being used sufficiently well for interview preparation, but the gap between expected and used factoids indicates that either interviewers could go further to investigate additional "fall back" questions in case their prepared questions are invalidated for some reason in the interview, or the expected factoids include some selections that interviewers are not comfortable with delivering in an interview. The fact that rejected factoid counts are relatively high indicates interviewers are making an effort to select quality information that will be useful in the interview.

The fact that minimums are not zero is a good sign that Intek is being used for all interviews. However, one factoid is probably not enough for effective questioning, with the exception of the Course topic as previously mentioned and the Home topic which is sometimes loaded purely to cross-reference its location with subsequent topics. We investigated these minimums further and found they are caused by several issues. Interviewers sometimes reject potentially good factoids, especially during their first Intek interview; a possible cause of this is that more technical searches, for example Python, required some technical knowledge to be able to differentiate good factoids from bad. Another issue with some early interviews is that interviewers would give up on a topic after finding one good factoid,

typically a summary. A final issue is that some entities contain little useful information for test questioning in their web presence, usually more well known companies such as Tenpin Bowling.

Maximums are generally comparable with expected, which indicates some interviews have been thoroughly prepared for. High maximum rejected factoids indicates both thorough preparation, but also that Intek may return a large amount of useless information: this is somewhat to be expected as Intek tries to handle the unpredicability of search terms by returning different types of information that is unlikely to all be relevant for a single search.

We now compare the quality of individual interviews using the same scoring mechanism used in the previous section.

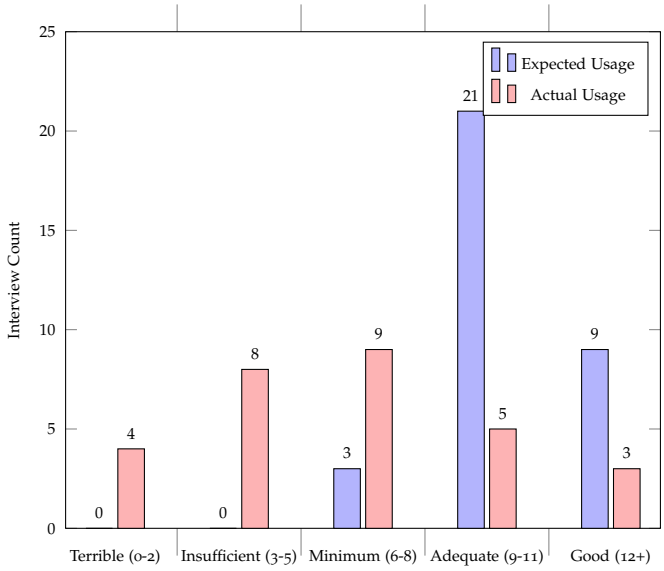

Figure 5.11 Counts of interviews by interview quality. Quality is scored by combining the number of Intek searches with the number of usable factoids returned. Each interview quality group is shown with its score range.

Figure 5.11 compares the expected Intek usage quality score from the previous section with interviewers' actual usage extracted from Intek logs. Note that actual interviews total only twenty-nine as four early Intek logs were lost due to an error that prevented the logging of the interview identifier. The same Intek usage quality score as in the previous section is used here for expected and actual, rewarding multiple searches per CV item with mutiple opened factoids up to a limit of three factoids per topic, which is seen as the minimum requirement for interviewing with CCE. This requirement aims to maximise cross-referecing between topics and to increase the diversity of test

questioning, for example interviewers were trained to use an Organisation, Role and Tool topic combination for a single job role item in a CV.

The comparison in Figure 5.11 reveals a large difference between expected and actual usage. This difference is somewhat expected given the differences between expected and actual in factoids returned in Figure 5.10. Investigations of individual interviews, especially those rated "Terrible" and "Insufficient" in Figure 5.11, reveal some causes which we list below.

Figure 5.9 showed less Job and Tool searches than expected, with these searches being omitted completely for some lower scoring interviews. This is unfortunate behaviour as Job and Tool are ideal topics for cross-referencing CV job roles. Investigation found several causes for this behaviour. In some cases CVs are sparse, containing few if any details which might be used for additional searches, this was the case for 50% of Terrible and Insufficient scored interviews. Where CV details do exist for further searches, they sometimes require confidence with Intek and an exploratory attitude to imagine which details might form useful searches, for example the name of a suspected technology tool hidden in a paragraph of explanatory text about a role. Where an interviewer has not received much useful information back from Intek, if they are trying exploratory searches and/or searches that prove difficult such as generic or obscure roles, tool or concepts, some interviewers may get discouraged at this point and give up on further searches for a CV item and move on, especially if they are short of time (which was frequently the case for interviewers in our study).

Figure 5.10 showed that interviewers are accepting/using less factoids than expected and are rejecting more. Investigation found many potential causes for this.

The expected scores are generated by an Intek expert for hypothetical interviews, whereas actual factoids are selected by an interviewer for use by themselves in a real interview. For this reason, some interviewers select less factoids, containing tests they are comfortable with, not necessarily the full range of expected probing tests encouraged by CCE which may potentially cause a more challenging interview suituation. Interviewer "comfort" is most likely to be the reason for factoid rejection where, in Figure 5.10, the total of used and rejected factoids combined are less than or close to the total of expected factoids, for instance Home, Organisation and Course topics. Where the used

and rejected combined total is much higher than expected, it is more likely that Intek is returning poor results that have been rejected due to low quality, for instance Job, Tool and Interest topics. This used and rejected split is used to analyse factoids type usage in Figure 5.12.

Another potential cause of low actual factoid usage is interviewers' inexperience with Intek and to a lesser extent interviewing, which might obscure the potential usefulness of some questions when planning in advance, whereas for expected we found that there are almost always some useful factoids to be found. The quantity of factoids returned by Intek can also add noise to the selection process, which can make spotting quality factoids more difficult. Also, interviewers may be following the initial CCE training advice to prepare two-to-five best test questions per CV item and are then moving on to the next item. This behaviour is amplified somewhat by the fact that a good factoid might contain multiple test questions. The Intek training encouraged interviewers to prepare more than five tests in order to give some "fall back" questions if the initial tests became invalidated for some reason, but interviews with quicker preparation times seem to have ignored this guidance. Further, interviewers may be deliberately closing worse factoids to "tidy" a topic before using it in an interview. Another alternative is that interviewers, perhaps when rushing, are not checking all factoids, but opening their favourites or those highlighted as relevant by the Intek heuristic. Also, one interviewer struggled with topic selection, chosing the incorrect topic type for a search, which consequently returned no useful factoids.

A final cause of a low actual factoid usage score, may lie in the scoring mechanism itself. The fact that investigations found *all* interviews covered at least two-thirds of CV items with a least one factoid, combined with the fact that multiple test questions may by contained in a single factoid, indicate our score may be implementing the CCE guidelines too harshly. We found that scores above 3.67 (Insufficient) were actually acceptable interviews, not fully cross-referenced with the ideal variety of tests suggested by CCE, but acceptably testing each CV item nontheless.

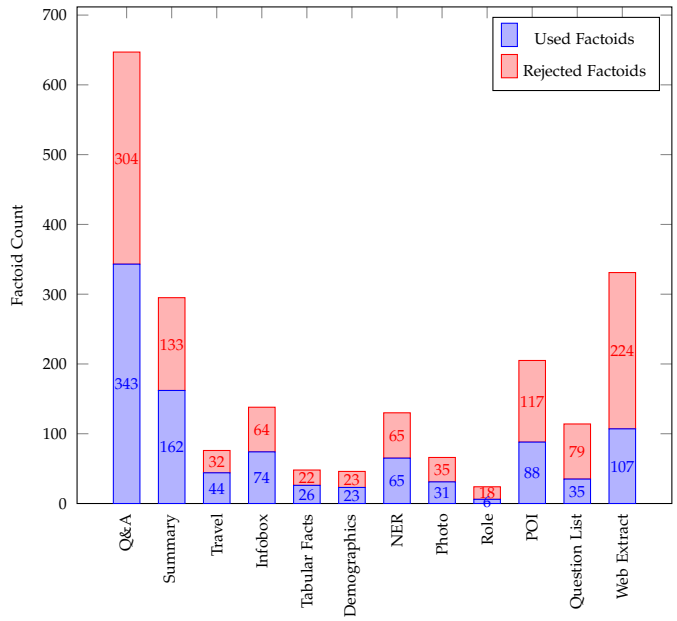

Figure 5.12 Factoid types used and rejected by interviewers over all interviews and topics. The most successful to least successful used to rejected ratios are shown left to right.

Figure 5.12 shows used and rejected factoids by the type of factoid, which allows us the examine the performance of these types of information. Note that the used and rejected factoids in each bar are *separate* factoids, the bar does not indicate that the used factoids in the bar have been rejected. Also, these factoid counts are for all interviews across the whole study in total. See Chapter 3 for more information about the implementation of these factoid types. Figure 5.12 is ordered by the rejected to used ratio; factoid types on the left might be considered better quality in general than those on the right. However, not all types of factoid are designed to operate well generally, some deliver info for specific types of searches, such as a web extract from smaller organisation web sites. We now discuss the performance of each factoid type in order.

Question and Answer (Q&A) is the most successful factoid type in terms of rejected to used ratio, which may be due to the use of the question as their title, which gives the interviewer a reasonable idea of the contents without opening the factoid. Q&As are sourced from frequently searched questions for a given search so can be very insightful, however they do have the greatest total of rejects which is probably due to their heavy use in Job, Tool and Interest topics, which leaves them open to issues of difficult generic and obscure searches as previously discussed.

Summary appears as the first factoid for all topics and is popular as a way of familiarising interviewers to the subject and apparently also as a source of test questions. Our technique of further summarising the initial summaries with NLP technology (see chapter 3) renders summaries shorter and more easily digestable.

Travel is popular as an unexpected test of commute routes that is cross-referenced between Home and work Organisation locations. This information was sometimes rejected when it transpired that interviewees lived in a different location to that stated on their CV, especially if in temporary student accomodation, this is an Intek issue for future developement.

Where an Infobox is available for larger Organisations, they usually contain multiple useful facts.

Tabular Facts, Demographics and NER People and Location extractions appear in fewer topics and are potentially very useful if they can be delivered in a context that seems relatively natural. For example, asking about the names of people an interviewee works with can be an excellent test, but must be carefully delivered. Conversely, the NER extractions were useless 50% of the time as they are completely dependent on this type of information being available in a organisations's web presence.

Photo can be an excellent source of various questions, for example intererviewers asked good unexpected tests about logos, signs and other objects near company premises. However, the photo is completely dependent on the availability and usefulness of the image in Google maps.

Role is potentially a source of tests about advertised Job roles, however this was apparently not used frequently, maybe as a consequence of the quality and popularity of summary factoids.

Points of Interest (POIs) are another excellent source of unexpected tests that require careful contextualisation to be delivered well, for example asking about favourite locations near an interviewees Home, requires a confident interviewer. POIs have clearly been made use of surprisingly often given they only appear in quantity in the Home topic, but with many rejections, probably due to the delivery difficulty.

Question Lists are "interview questions" or "quiz questions" sourced from web search and appearing last in a topic are considered something of a wildcard or fallback if other factoids have failed to deliver.

It appears that mostly these questions are not relevant or inferior to other factoids, but have been used approximately once per interview.

Web Extracts deliver summary information extracted from the landing page of company web sites and are aimed at providing information for the smallest organisations that are absent from Wikipedia or Google Maps. These factoids are not designed to be relevant for larger companies which appears to be the case as they are rejected in two-thirds of searches.

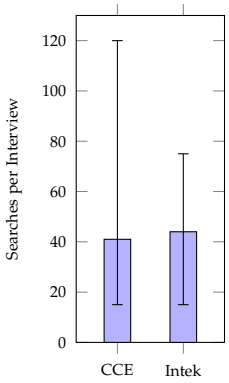

Figure 5.13 The mean, minimum and maximum interview preparation times in minutes for each condition.

Figure 5.13 shows the mean, minimum and maximum interview preparation times reported by interviewers in the post-interview questionnaire.

Comparing interview preparation times across conditions, Intek interviews take three minutes longer to prepare for on average, which is not a significant difference to CCE and is only one minute short of the expected average preparation time from the expert Intek interview preparations. The only significant difference in preparation times across conditions is the reduced maximum time for Intek, which indicates a more stable, standardised preparation with more uniform quality.

In the Intek condition, individual interviewers were very consistent with their preparation times, with an interviewer preparation variability of ten minutes or less (with the exception of one interview), indicating interviewer attitude is the major factor in preparation times overall, rather than CV or Intek quality. We find overall that interviewers using a medium amount of time to prepare scored better using our Intek usage quality score (43-47 mins preparation performed best) and detected deception more often (33 to 60 mins preparation performed best) with quicker and slower preparations performing

worse. These results indicate again that interviewers with more skill and confidence using Intek peform better.

The results presented in this section lead us to believe that Intek usage quality was good overall and in all cases covered the majority of highlighted CV items with at least one test question.

The high-level results for Intek show a two-fold improvement in deception detection, yet our Intek usage quality scores do not correlate directly with deception detection success at the individual interview level. This indicates factors other than Intek are involved in deception detection success. This section has shown that Intek appears to contribute to test questioning. In the next section, we use the various data we have gathered to further support Intek's contribution to test questioning and two additional interviewer factors: control of the interview and behaviour judgement.

We do not explore factors in deception detection that Intek has no control over, such as interviewee or environmental factors.

5.2.5 *Contribution to Deception Detection*

In this section we examine Intek's contribution to the three main interviewer factors in deception detection: control of the interview; test questioning; behaviour judgement. These three interviewer factors have been identified through analysis of the CCE and Intek requirements, process and results. In this section we use data from interview transcript analysis, expert interviewer ratings of interview quality and questionnaires completed by interviewers and interviewees.

The first essential factor we analyse is control of the interview. Interview control is essential to manipulate the interview context such that test questions can be delivered, for all highlighted CV items, without the interview appearing as a strange interrogation. Control can be difficult to achieve if the interviewee is talkative, offering information in advance, and also if the interviewer is a talkative type who might struggle maintaining a professional distance. We expect Intek to have a moderate impact on control due to the prepared interview offering a "script" to manage coverage of highlighted CV items and giving a sense of context within each topic.

The next three figures present interviewers' perceptions of their interview control using questionnaire ratings on a seven-point Likert scale.

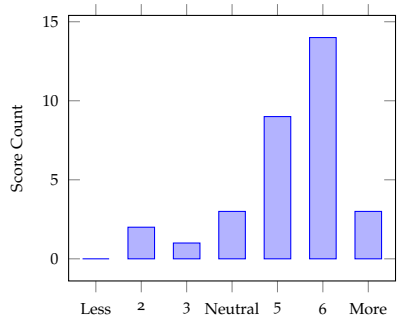

Figure 5.14 Interviewer post-interview response to the question: Compared to CCE with no app, how in control of the interview did you feel using the app?

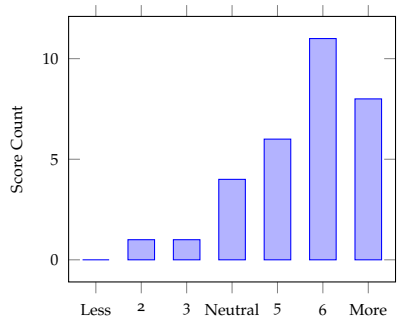

Figure 5.15 Interviewer post-interview response to the question: How did the app affect your ability to deal with topics that were new to you?

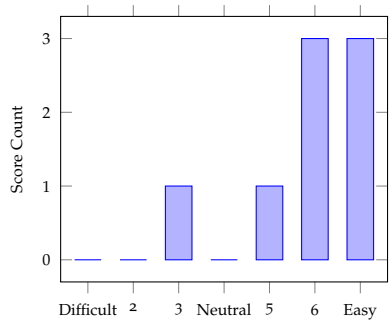

Figure 5.16 Interviewer post-study response to the question: Overall, how did you find these tasks using the Intek application: - Using this script for questioning during the interview?

Figure 5.14 shows clearly that interviewers felt more in control of the interview as a whole using Intek. Figure 5.15 shows clearly that interviewers felt they were more able to control discussions around topics they previously knew nothing about. As our interviewers were fairly young novices, that were at times intimidated by more experienced interviewees, this is an especially important measure. Figure

5.16 shows that interviewers, with only one exception, found using the preparared interview "script" in Intek easy to use for questioning. This indicates that the Intek script is likely a direct cause of the increased perception of control in these three figures.

We now examine the more objective expert ratings of interview performance around interview control.

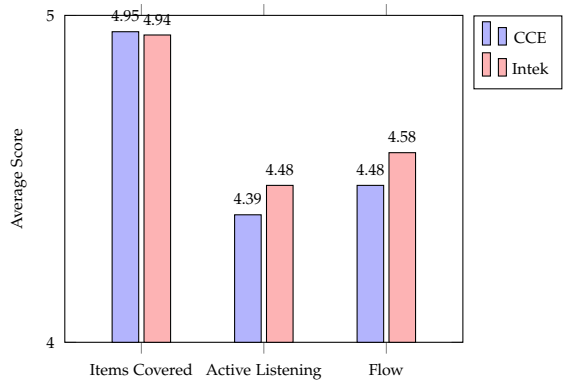

Figure 5.17 Comparison of average scores between CCE and Intek for control-related factors rated by an expert interviewer. All scores are out of five. Due to the high scoring regime, we have zoomed-in to the four-to-five score range.

Figure 5.17 shows the averages of three scores marked out of five related to interview control quality. These scores were produced by an expert interviewer after reviewing recorded interview videos showing both participants simultaneously. The scores given are all high, with 95% scoring four or five, which may have been done to avoid appearing overly harsh to interviewers; consequently the score differences we present are likely proportionally more significant than they appear.

Items Covered indicates simply that the interviewer covered all highlighted CV items in discussion. This is a measure of basic interviewer competance, but does require the interviewer to be in control of the interviewee and aware of interview progress overall. As we would hope, a very high proportion of interviewers covered all items, with an almost identical score between CCE and Intek.

Active Listening is an interviewer skill and method of control that requires careful observation and feedback, with the aim of gathering information, observing behaviour and manipulating discussion into areas of enquiry. Intek average scores for active listening are slightly higher than CCE, perhaps indicating the small extent to which interviewer experience has improved performance between conditions.

Flow is a measure of how smoothly the interview transitioned between topics and questioning, which requires the interviewer to have a good sense of their progress through the interview overall as well as the completeness of their questioning for particular items. Intek average scores for Flow are slightly higher than CCE, which may indicate the impact of interviewer experience or the infuence of the Intek script.

Summing up interview control, we find interviewer perception of control with Intek is definitely improved over CCE and that the Intek prepared script functioned well in questioning, making Intek itself a likely cause in this improvement. Expert scoring from interview observation found small increases for Intek in two out of three factors in interview control, probably indicating the improvement in interviewer experience and skills as well as the assistance of Intek.

The second essential factor we analyse is test questioning. A good breadth and depth of test questioning for each topic is essential in order to elicit behaviour change in deceivers. The generation of information for test questioning is the primary function of Intek, which due to a fact-based approach, has the added benefit of veracity testable questioning. Consequently, we expect Intek to have a high positive impact on test questioning.

The next three figures present interviewers' perceptions of the quality and quantity of Intek information and the difficult of extracting tests from this information.

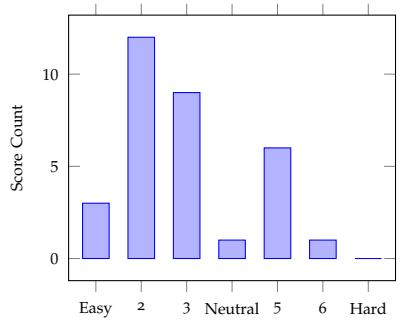

Figure 5.18 Interviewer post-interview responses to the question: How did you find the task of extracting useful tests from the "info items" supplied by Intek?

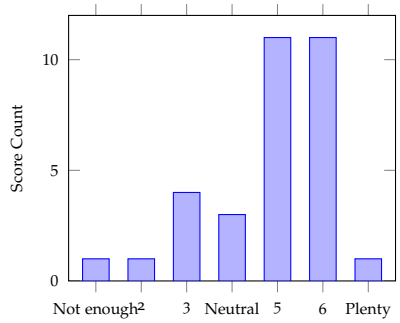

Figure 5.19 Interviewer post-interview responses to the question: Could you find enough satisfactory tests?

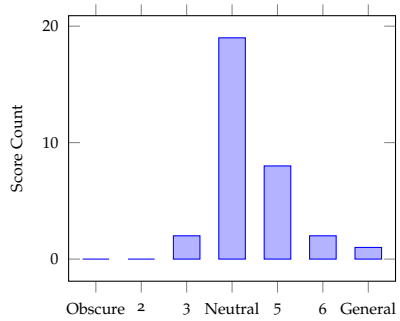

Figure 5.20 Interviewer post-interview responses to the question: How was the "episodic-ness" of the info returned?

Figures 5.18 and 5.19 show that, overall, interviewers found it easy to extract tests from the information supplied by Intek and that there was sufficient information available for their test questioning. This is an important finding as the previous section shows that Intek delivers apparently useful tests, but this finding confirms that interviewers are able to easily extract test questions from this information.

Figures 5.18 and 5.19 show a few interviewers found extracting tests harder or the quantity of tests insufficient. The reasons reported for this were lack of good information when searching for very generic terms or very niche terms, or incorrect information if search terms were ambiguous or old. Some interviewers also struggled to deliver specific, seemingly obscure, test questions, especially when interviewing more experienced interviewees.

Figure 5.20 shows that the "episodic-ness", the level of obscurity or depth of Intek information that allows it to effectively probe rich episodic accounts, is, overall, pitched at just the right level, with a few exceptions toward to the more general, causes for which are mentioned in the previous paragraph.

We now examine linguistic statistics extracted from interview transcripts and a single question from interviewee questionnaires.

Condition	Interviewer Word Count	Interviewee Word Count	Total Word Count	Total Question Count
CCE	706	2684	3390	28.91
Intek	776	3588	4364	33.33
Increase%	10%	34%	29%	15%

Table 5.5: Transcript statistics from CCE and Intek conditions. The percentage increase for Intek over CCE is also shown.

Condition	Deceiver	Truth-Teller
CCE	3.1	2.8
Intek	3.5	2.0

Table 5.6: Interviewee post-interview responses to the question: How challenging was the interview? The average of seven-point Likert scores are shown for CCE and Intek conditions grouped by truth-teller or deceiver.

Table 5.5 shows everyone spoke more per interview; 29% more words in total with 15% more questions being asked by interviewers, likely prompted by the availability of good questions supplied by Intek.

One of the aims of the CCE method is that interviewers should speak sparingly, using good questions, with further prompts, to encourage deceivers to do most of the talking thereby expanding on their lies, increasing cognitive load for them. The episodic nature of questions should conversely make questioning an easy process for truth-tellers. Table 5.5 shows interviewers spoke 10% more asking 15% more questions, while interviewees spoke 34% more, which indicates interviewers using Intek are asking better questions, which is encouraging more interviewee talk. The results of this improved test questioning with Intek are shown in Table 5.6, which shows interviewees' perception of interview difficulty, demonstrating exactly the desired outcome, of significantly increased challenge to deceivers, with significantly decreased challenge to truth-tellers.

We now examine the expert ratings of interview performance around test questioning.

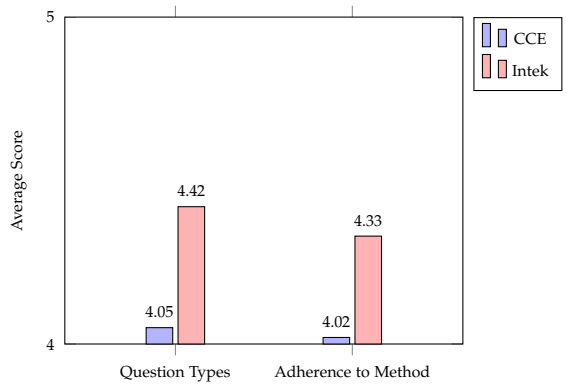

Figure 5.21 Comparison of average scores between CCE and Intek for test-questioning-related factors rated by an expert interviewer. All scores are out of five. Due to the high scoring regime, we have zoomed-in to the four-to-five score range.

Figure 5.21 shows the averages of two scores marked out of five related to test questioning quality. These scores were produced by an expert interviewer after reviewing recorded interview videos showing both participants simultaneously. As with Figure 5.17 the scores given are all high, with 99% scoring three or above, therefore the effect shown may be greater than it appears.

Question Types indicates the quality and quantity of test questioning. Adherence to Method indicates the interviewer has correctly used CCE methods, the most important of which is test questioning, but also includes the good contextual delivery of those tests. A significant improvement is shown for Intek over both of these scores, objectively showing the quality and quantity of available Intek information, but also indicating that this information is being effectively transformed into test questions and delivered well.

Summing up test questioning, we find a significant improvement for Intek over CCE in interviewer and interviewee perceptions, transcript statistics and expert scoring.

The third and final essential factor we analyse is the judgement of interviewee behaviour change by interviewers. Good interviewer judgement requires a baseline of "normal" behaviour, good observation of behaviour throughout the interview and good judgement to discern deceptive behaviour from normal. Behaviour judgement is the primary contributor to interviewer suspicion of deception, the secondary being veracity checking. We do not expect Intek to directly affect judgement, other than to make more time for observation by improving the efficiency of the previous factors and creating more

opportunties to observe behaviour change through the use of good test questioning.

The next two figures present interviewers' perceptions of the time they had to observe and how demanding the interview was overall.

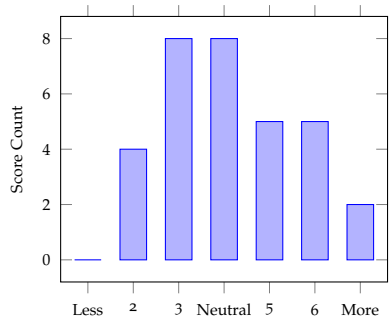

Figure 5.22 Interviewer post-interview responses to the question: How did Intek affect how much time you had to observe?

Figure 5.22 shows that interviewers' perception of the time they had available to observe behaviour using Intek was mixed. Those that perceived they had more time available reported increased confidence through familiarisation with topics and by following the interview "script". Those that felt they had less time available, did not report many issues, but those that were reported included difficulty simultaneously managing the multiple tasks of interviewing and lack of useful information retrieved for very generic job titles.

Condition	Average Score
CCE	4.0
Intek	3.5

Table 5.7: Interviewer post-interview responses to the question: How mentally demanding did you find this interview? The average of seven-point Likert scores are shown for CCE and Intek conditions..

Table 5.7 shows a significant decrease in interviewers' perception of how demanding interviews were using Intek.

Given that interviewee experience levels, a typical factor in interview difficulty reported by interviewers, are similar between CCE and Intek (see Figure 5.2) and that environmental factors, such as bad internet connections, are evenly distributed across CCE and Intek conditions, we can assume less demanding interviewing is due to improved interviewer factors: control of the interview and test questioning, which we have shown have been significantly improved by Intek.

Summing up this section, we have shown there is good evidence for Intek's contribution to two essential factors in deception detection: interview control and test questioning. Judgement of deceptive behaviour appears to have been positively affected indirectly by Intek through the previous two factors. We believe this is as far as we can go in proving Intek's contribution to deception detection. We now move on to the final section and review interviewers' perception of Intek's usability and the main reported issues with Intek.

5.2.6 *Usability*

In this section we examine how well Intek was accepted by users by analysing feedback given in interviewer post-interview and post-study questionnaires to build a picture of Intek usability.

The system usability scale (SUS) (Brooke et al., 1996) was developed as a tool to quickly assess the usability of a product or web site. It has now become an industry standard, having been used in over a thousand publications. SUS uses ten five-point Likert scale questions which deliver a single final score (out of 100, but not a percentage) which is easily compared with existing systems. Bangor et al. (2008) examined ten years of SUS studies to establish a mean score of 70.14 which in general indicates the bare minimum for a passable product, with better products scoring 80 to 90 and superior products greater than 90. Intek achieved a "better" mean score of 84.4, with some variability in scoring discussed below. SUS scores should be discussed in context with the overall task success rates and any issues that arose. We report a two-fold improvement in deception detection for Intek over baseline (see high-level results Section 5.2.2) and we discuss Intek issues and solutions later in this section.

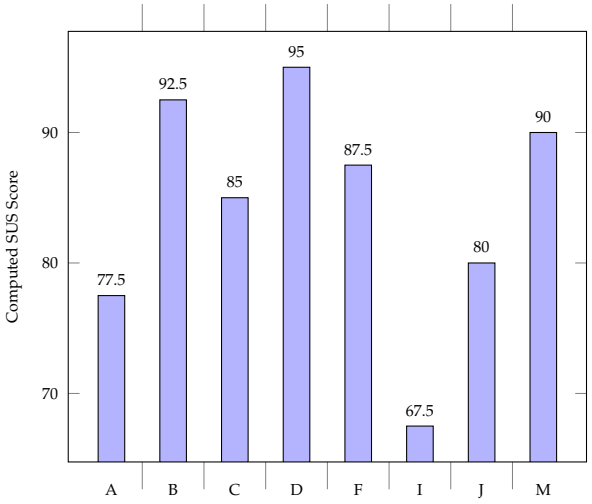

Figure 5.23 Computed system usability scale (SUS) scores for interviewers that participated in the post-study questionnaire.

Figure 5.23 shows SUS scores for the interviewers that participated in the post-study questionnaire. All scores are in the "better" range with the exception of interviewer I at 67.5 and interviewer A at 77.5.

Interviewer I raised some valid issues with Intek including the previously mentioned "multiple homes" problem which can invalidate future topics' cross-referencing and also a lack of useful information returned for generic job titles. Interviewer I only conducted one Intek interview, so inexperience may have added to their frustrations, but they were equally complementary, mentioning that Intek gave them ideas they would never have thought of and that all information was presented "in the right place".

Interviewer A gave a fairly good, but not "better", SUS score and reported liking the "good order of events to work by". Interviewer A conduted five Intek interviews, but did not report any Intek-specific negative issues, rather a general issue about the difficulty of separating "bad but truthful" answers from lies.

Intek issues mentioned by interviewers are discussed later in this section with possible solutions.

The next two figures present questionnaire results for Intek satisfaction and enjoyment.

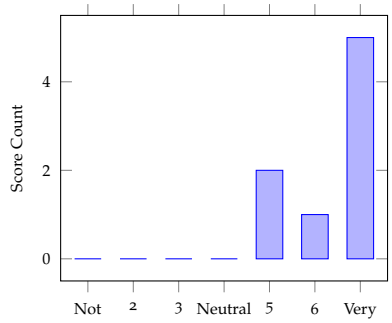

Figure 5.24 Interviewer post-study response to the question: Overall, how satisfied are you with Intek?

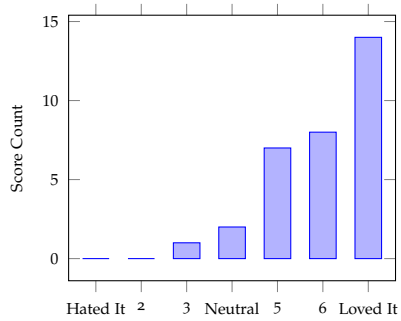

Figure 5.25 Interviewer post-interview responses to the question: Did you enjoy using the app in general?

Figures 5.24 and 5.25 show that post-interview, in "hot blood", and post-study, with the benefit of hindsight, and despite the two lower SUS scores, all participating interviewers were satisfied with Intek and all but three interviews enjoyed using Intek. The only interview in which the interviewer did not enjoy using Intek, was interviewer I that gave the lowest SUS score above.

The next two figures present questionnaire results for Intek ease and effectiveness.

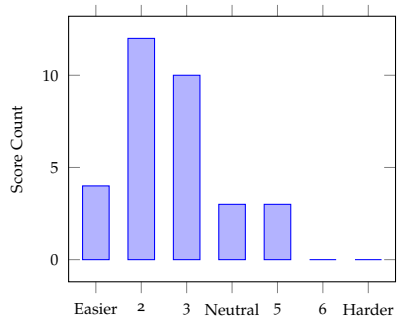

Figure 5.26 Interviewer post-interview responses to the question: Compared to CCE without Intek, how *easy* did you find the task of detecting deception with Intek?

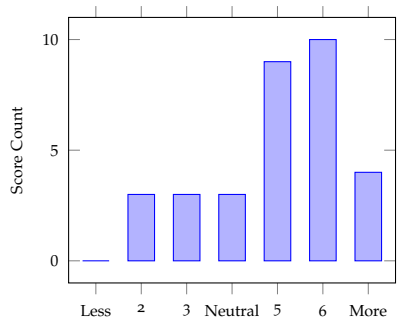

Figure 5.27 Interviewer post-interview responses to the question: Compared to CCE without Intek, how *effective* did you find the task of detecting deception with Intek?

Figures 5.26 and 5.27 show that, post-interview, interviewers found interviewing with Intek significantly easier and more effective than using CCE alone. Ease is an important result as it reinforces the view that transforming Intek information into test questions and navigating the Intek "script" in real-time is not a problem, even for novice interviewers. A positive effectiveness rating is in line with Intek's high-level results and the discussed contribution to interview factors in deception detection.

Issues raised by interviewers in the three interviews that found Intek harder to use (in Figure 5.26) were: an overall difficulty dealing simultaneously with all interviewing tasks while using Intek and Zoom; ambiguous searches not being flagged up effectively; difficulty selecting the correct topic types.

Issues raised by interviewers in the six interviews that found Intek less effective (in Figure 5.27) were all related to Intek not extracting enough (or any) useful information. The reasons for these failures were poor extraction from existing web pages, the Home topic not extracting useful information about other locations and incorrect or missing information extracted from out of date web pages.

The final two figures present questionnaire results for the tasks of preparation in advance of the interview and using Intek for real-time search during the interview.

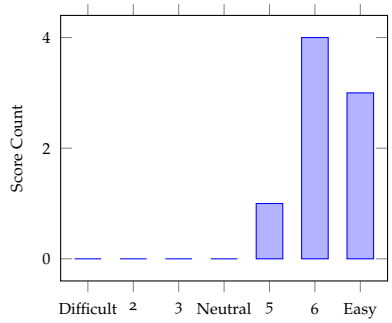

Figure 5.28 Interviewer post-study response to the question: Overall, how did you find these tasks using the Intek application: - Using the system to prepare an interview "script" before the interview?

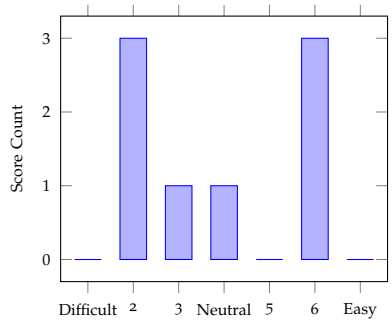

Figure 5.29 Interviewer post-study response to the question: Overall, how did you find these tasks using the Intek application: - Using the system to search for new topics during the interview?

So far we have discussed Intek search as a homogeneous task and indeed Intek does function in the same way whether it is used for preparation *before* an interview or real-time search *during* an interview. However, Intek was designed with real-time search use in mind and interviewers were trained in using Intek in response to interviewee dialogue in real-time. Real-time search is especially useful if the interview seriously diverges from the prepared material. Real-time search might also be useful if the information available before the interview is sparse, as with some of our CVs, or missing altogether, as with other interviewing scenarios such as security. In this job interviewing study there is not a large incentive for interviewers to use real-time search as they have at least a reasonable preparation ready before the interview and have much to concentrate on as it is, especially as novices. Therefore, we do not expect many interviewers to make use of real-time search.

Figure 5.28 shows that post-study, participating interviewers were happy with the preparation aspect of Intek, reporting that prepara-

tion was made signifcantly easier by the centralisation and diversity of information.

Figure 5.29 shows that most interviewers found real-time search difficult, which is understandable as interviewing is a difficult task without simultaneous search. However, three interviewers embraced real-time search, finding it relatively easy. These three interviewers mainly used the reliable Tool topic to search for software packages that were mentioned during the interview to "come up with further questions to test knowledge" and allow "off the cuff" testing.

A separate study in security interviewing or other scenario without interviewee information supplied beforehand would be required to test real-time search fully. Some of the learnings from this study would be of use in optimising Intek further for real-time use, for example better indications or prioritisation of the most relevant information for a topic.

We now move on to list important issues that have been reported with Intek and briefly discuss their impact on this study and future work. Many of these issues have been mentioned previously under the relevant sections.

The most important issue raised is the lack of useful information returned in certain circumstances, which has clearly impacted the effectiveness of Intek in some cases and in a few cases misled the interviewer with (currently) false information. For generic searches, especially Jobs or Interests, for example "volunteer", this is a difficult issue to fix. This is why Intek offers multiple cross-referenced topics to supply further relevant information for a single CV item. Larger companies, for example Caffe Nero, that have multiple branches, require a little more work from the interviewer with an additional search specifying the branch and if that is still ambiguous specifying the location, for example Caffe Nero Western Road Brighton. In future work, nearby alternatives could be offered to the user. For obscure or niche searches, for example "medical electronics" or "world youth coral championships", Intek fails to extract pertinent information even though a suitable web page is available, due to the diversity of the information. Intek wrappers should be improved to make a better job of extracting useful information or summarised paragraphs from these pages. If a search has no web presence to extract from, this should be presented better to the user. Finally, the age of the source web page should be reported if possible, to indicate the presence of potentially out of date information.

An issue with multiple homes was reported by multiple interviewers. The issue is caused by interviewees who do not currently reside at the Home location reported in their CV. These are mainly students, but this situation could apply in multiple scenarios. In these cases, the Home search itself is still valid, where highlighted as a discussion item, as the interviewee should still have deep knowledge of that location. The problem is that some of the most useful cross-referenced factoids in subsequent topics that rely on a current Home location are invalidated. A fix for this would involve gathering updated Home details during the interview and adding the ability to toggle the original Home and the new Home with regard to cross-referenced information in other topics.

Multiple interviewers reported that information regarding incorrect Organisations with ambiguous names might be returned by Intek and accidentally mistaken for the intended Organisation. This occurs with the Organisation search which integrates multiple data sources, any of which could be based on incorrect entities. Intek already tries to flag this up in the topic box, as well as displaying individual company names on factoid titles returned by each data source, but these indications might be made more obvious to the user.

One interviewer reported that the Home topic does not support non-home locations, such as specific locations abroad for events. Intek would benefit from the removal of some of the lesser used nearby-location Home information and the introduction of more familiarisation-based location information to handle general locations not so well known to an interviewer.

Two issues were raised in which interviewers were acting counter to the training session, indicating that training should emphasise these points more effectively. Firstly, one interviewer struggled with topic type selection, but this is not always 100% obvious, some unexpected searches may require some exploration to find the correct topic type. Secondly, one interviewer struggled using Intek, Zoom and the CV simultaneously with the other interviewing tasks. Training recommended Intek replaces the CV, however this interviewer was not comfortable with this, causing them difficulty switching between Intek, CV and Zoom.

To conclude this section on a positive note, we present interviewer qualitative feedback on aspects of Intek multiple interviewers found particularly useful.

Interviewers found the overall task of preparation easier: "when googling it can be very overwhelming trying to sift through for relevant info but the app does that for you which I'm grateful for".

Interviewers reported finding the user interface "easy to navigate throughout the interview" as "you have the data condensed in one little convenient page". Additionally, "it was a lot less demanding as I was able to just scroll through and see what I've highlighted and ask accordingly".

Interviewers found Intek's test questioning support useful: "it gave me good challenge questions to ask, the pictures the app provides are so helpful and I find great challenge questions that I would've never thought of", as well as "I don't have to try to come up with tests along the way" and "it helped inform questions where I was stuck".

Interviewers reported understanding unfamiliar topics better: "I didn't know a lot of the companies they worked for so it was really useful to get an idea of what they did before asking them, rather than going in blind and being completely overwhelmed" and also "I felt quite out of my depths with the jobs highlighted and the app made it a lot easier to understand what their role was and what the companies that they'd worked for were".

Interviewers appreciated the ability to quickly veracity check answers as they "had the answers right there to check". Intek's factual information helped verify topics interviewers were unfamiliar with "I can verify information which otherwise would be easy to deceive me" and also gave interviewers an opportunity to interview more strategically "I was able to fact check while processing his answers, it also informed me where I should press harder if I ever suspected he was lying".

With that we conclude the presentation of our job interviewing study, results and analysis.

5.3 CHAPTER SUMMARY

In this chapter we present high-level Intek results which significantly improve over two baselines. We demonstrate that Intek was used in all interviews at least adequately and in most interviews well and that this usage made a significant contribution to the essential interviewer factors in deception detection. We then present positive results for the usability of Intek, while finally discussing issues raised by interviewers with some potential solutions.

6

CONCLUSION

This chapter summarises the thesis as a whole, highlights the main contributions of this work, lists some limitations of our work and provides possible directions for related future work.

6.1 SUMMARY OF THESIS

In summary, Chapter 2 contextualised the CCE and Intek approach to interviewing for deception detection by examining key related work concerning interviewing methods and the various factors used for deception detection. We then put Intek in context by reviewing technological methods that have been implemented for deception detection. We then move on to provide background in methods for information extraction and named entity recognition, both of which are required to fully contextualise our approach to web NER, which spans IE and NER. We then examine free-text segmentation methods and HTML tag representations, both of which are directly related to our design decisions for web NER.

Chapter 3 deals with the entire lifecycle of designing and implementing Intek. We initially give an overview of the CCE method on which Intek is based and then describe how the key elements of CCE relate to Intek. This gives an insight into the motivations behind Intek design. We then describe the iterative development lifecycle we use to centre our design and development on stakeholder input, using quick design, prototyping and feedback iterations. We then give background and justification for key high-level design decisions that shaped Intek to a large extent. We then give detail on every step in the lifecycle as it progressed: initial requirement gathering and analysis, the design principles applied, the methodology and overview of key elements and finally evaluation. We split this detail into the two main sections of Intek work: the front-end user interface, and back-end information extraction.

While most of the underlying NLP technology we used in Chaper 3 was "off the shelf", we discovered an opportunity to investigate a novel approach to web named entity recognition using HTML tags.

Chapter 4 details our work in this area. We introduce our method and explain our motivations for this investigation. We then go on to describe the datasets, models and techniques we used to evaulate our approach. Finally we discuss the results, how our method produced these results and how our technique might be applied in future.

Chapter 5 is concerned with the evaluation of Intek. We begin by describing the participants, conditions and overall process of our study, which involved 111 interviewees, split into three conditions, interviewed by 13 interviewers, over the course of four months. We then go on to describe the various sources of data collected during our study, that are used to generate our results. We present our high-level results. We then go on to discuss our other results in an order that demonstrates Intek's contribution to deception detection. Firstly, whether Intek was capable of returning useful information for all interviewees in the hands of an expert. Secondly, how real interviewer usage compared to expert usage. Thirdly, how this usage corresponded to the essential interviewer skills required for deception detection. We also present a usability analysis of Intek with reported issues and suggested solutions.

6.2 MAIN CONTRIBUTIONS

The main contributions of our work in this thesis are split into sections defined by the those communities that might benefit: researchers in the field of deception detection; developers of interviewer support tools; researchers in the field of natural language processing; practitioners of information extraction in industry.

6.2.1 *Deception Detection Researchers*

We have shown high-level results for Intek that represent a two-fold increase in deception detection performance over our baseline approaches. Intek gives accuracy in the top 1% of studies in Bond Jr and DePaulo (2006). Furthermore, this result was attained in a realistic job interviewing scenario, using novice interviewers without specialised contextual knowledge in the topics discussed. The CCE method appears only to perform well in a job interviewing scenario with Intek support.

In our results we have shown Intek performed well (Section 5.2.3) and was used well in most cases and at least adequately in all cases

(Section 5.2.4). We have also shown that the information provided by Intek has significantly improved the effectiveness of test questioning, resulting in more talkative interviewees and increased cognitive load in deceivers (Section 5.2.5). This gives us confidence that the data we have gathered around Intek is related to a genuine positive effect on deception detection in interviewing, and enables us to draw some conclusions in this field.

Our first conclusion is that a little expertise improves deception detection performance and this expertise can be gained quickly using a tool such as Intek. We now explain this point more fully.

CCE and Intek use the same overall method for test questioning, in that questions should be unexpected tests of expected knowledge, near-episodic and veracity-testable. Intek adds some level of interviewer expertise in each topic through summaries and a good range of information, quickly accessible in a predictable layout. Through presenting factual information, Intek also allows quick veracity testing of answers. Intek's "quick expertise" and veracity testable approach are major contibutors to Intek's success in deception detection.

Analysis of interviewer feedback shows us that this basic topic expertise had a significant positive impact on interviewers. In results Section 5.2.5 Figure 5.15 we show that in 80% of interviews interviewers were confidently dealing with new topics. In Section 5.2.4 Figure 5.12 the Summary factoid type is one of the most successful, showing this type of familiarisation information was frequently used by interviewers. Additionally, the majority of interviewer qualitative feedback mentioned the ability to deal well with new topics, for example "It was much easier to create test questions because you understand the subject better", "I was able to ask more specific questions particularly about topics I am very unfamiliar with" and "I didn't know a lot of the companies they worked for so it was really useful to get an idea of what they did before asking them, rather than going in blind and being completely overwhelmed". It appears that basic topic expertise allows interviewers to recognise the most effective diagnostic tests to pick, which in turn further increases their topic knowledge. Also, if the interview discussion moves away from prepared material, interviewers have enough background knowledge to at least assess the plausibility of answers.

The benefits of topic expertise are shown in literature Section 2.2.2 and we believe our findings lend support to theories of contextual ex-

pertise and diagnostic questioning. We conclude that Intek supports a contextual expertise approach to deception detection.

Our second conclusion is that the perceptual and contextual information delivered by Intek appears to be more successful in deception detection, supporting reality monitoring theory. We now explain this point further.

In results Section 5.2.4 Figure 5.12 we show factoid types that supply contextual and perceptual information are the most successful by ratio of used to rejected and are also the most frequently used. These factoids types are: Q&A; Summary; Travel; Infobox. Also, interviewer qualitative feedback indicates organisation logos from infobox, various details from photos and route-based travel information are particularly useful.

The reality monitoring literature in Section 2.2.3 states that the presence of primarily contextual and perceptual information is indicative of truth. The fact that Intek interviewers make good use of these types of information should place additional cognitive load on decievers as they are forced to manufacture experiential information in real-time. We conclude that Intek's successful deception detection result using these types of information lends support to reality monitoring theory.

6.2.2 *Interviewer Support Tool Developers*

The further development of Intek may be of interest to researchers in interviewer support tasks in many scenarios, as well as candidate interviewing practitioners in industry.

We have developed a successful practical interviewing tool for deception detection, which is, to our knowledge, the first of its kind. Our development process and analysis of feedback has given us insight into task performance, topic selection, extraction and presentation techniques. We now present key insights and design implications in these areas.

We have analysed the performance of "sub-tasks" to indicate where future work might best be focussed. In results Section 5.2.5 we examine the control, test questioning and judgement aspects of interviewer performance with Intek. We conclude that control and test questioning are well supported, whereas we may look to enhance support of the judgement aspect of interviewing by implementing a complimentary technology such as real-time linguistic analysis (see future work Section 6.4). In results Section 5.2.6 Figure 5.29 we examined the

use of Intek real-time search during interviews. We found real-time search was used in a few interviews, but it is not clear how effective this would be in a real-time only scenario, such as security interviewing. This is likely an area for a future study, using a leaner Intek, in which preparation materials are not available.

We found our selection of six topics worked well. Less than six would have curtailed exploratory searches, while more than six may have confused interviewers. The only potentially missing topic, which occured multiple times in interviewer feedback, was to handle events, such as festivals or sports matches.

With regard to Intek data sources and extractors, all played a part and indeed they were designed to be mutually supportive in the face of unpredictable searches. Data sources and extractors must be considered carefully for inclusion: do they supply data at a useful level of detail; how can the data be transformed to be consumed effectively. In results Section 5.2.4 Figure 5.12 we show some of the simplest factoids to implement are the most successful: Q&A; Summary; Infobox. Whereas the more complex factoids, the NER pipeline for example, was moderately successful. In the future we would not remove any factoids, as they are mutually supportive, rather we would aim to indicate the potentially most effective factoids to the user (see future work Section 6.4). In results Section 5.2.6 we identify the main Intek issues raised by interviewers and give potential solutions.

With regard to the Intek user interface; no UI problems were reported by interviewers (that followed the training advice) and results Section 5.2.6 presents a "better" SUS score of 84.4 out of 100 along with good results for ease and effectiveness. The key UI contributions to Intek's success are: a straightforward design philosophy; fixed layout for each topic, even if no information is returned; horizontal stacking of topics which represents the interview flow; neatly encapsulated red-bordered factoid groups within topics, that maintain group cohesion when the factoids within are opened or closed. In the future we aim to use the factoid relevance data gathered through the UI to improve the indication of the most effective factoids to the user (see future work Section 6.4). Finally, we were intially concerned interviewers would struggle to transform the different text, image, graph and table-based factoid information into usable tests, which would have been a major problem. It appears from results Section 5.2.5 Figure 5.18 and from qualitative interviewer feedback that, aside from problems with a lack of extracted information in some cases, that

transforming Intek information into questions is second nature to interviewers and not a problem at all.

Interviewers have to perform multiple tasks simultaneously, including using Intek, therefore the UI must be as simple as possible, including as few factoids per topic as possible. Conversely, a range of factoids of different types should be included to increase the cognitive load of an interviewee, mitigate against any factoids that fail to return useful information and provide some factoids containing basic fact-based familiarisation for the interviewer. The implication of this is that designers must strike a balance between simplicity and providing a useful range of factoids in the face of unpredictable search terms. Additionally, factoids that contain the name of the entity or concept being searched for is encouraged, as this gives interviewers the ability to quickly assess whether the factoid has returned information about the correct entity; questioning during an interview using information concerning an incorrect entity may lead to confusion, mistrust in the system and possibly a false impression of deception.

Developers interested in pursuing a re-active fact-checking approach to deception detection, similar to that discussed in Section 3.4.2, may be able to use our range of extractors as a basis for an "episodic" search. This web of data sources could eventually be extended to provide a generally useful source of personal information.

We have gathered substantial data in application logs, expert ratings, questionnaires, videos and transcripts that might be of use to both of the above parties.

6.2.3 *Natural Language Processing Researchers*

Our approach for the inclusion of HTML in NER from web pages might benefit any web-sourced NLP task and thereby be of interest to the NLP research community.

Our technique compares Text+Tags sentences with their Text-Only equivalents, over five separate datasets, using two NER models. We found increased F1 performance for Text+Tags over all datasets and models. These performance increases appeared to be due to HTML tags delimiting entities, which is discussed further in Chapter 4 Section 4.3.2. This delimiting effect is higher in tag dense web pages: we saw a performance increase of F1 13.2% in a particularly tag dense dataset, while a dataset containing more natural language containing sentences with fewer tags saw an improvement of F1 0.9%. We found

that pages with a higher tag density correlate fairly well with higher performance gains (Pearson correlation coefficient of 0.72). We conclude that this is good evidence for the inclusion of HTML in this way in other NLP tasks (see future work Section 6.4).

Our technique uses a simple one token per HTML tag representation. This representation of tags could be expanded to include information from HTML structure or visual style information (see future work Section 6.4).

6.2.4 *Information Extraction Industry Practitioners*

Our approach allows industry practitioners that might be interested in extracting Competitive Intelligence or Business Intelligence information, for example regarding competitor products and prices, to use a single technique for all web pages. Our approach can seamlessly perform NER on pages that contain any level of HTML, from HTML-dense repetitive record structures, that have previously been the domain of wrapper methods, to free-text sentences with minimal or no HTML. We find performance over text-only is increased most in the former case, as discussed in Chapter 4 Section 4.3.1. Our approach requires minimal extra pre-processing (to extract free text areas from web pages) and between 3% and 11% extra training time (to accommodate the extra tokens that represent HTML tags).

6.3 LIMITATIONS

We have identified three potential limitations of our work in this thesis. We now discuss their validity and potential solutions.

The first limitation is the degree to which interviewer skill and experience was responsible for Intek's improvement over CCE. Intek was the only major change between CCE and Intek conditions. While interviewer skill increased 3% between CCE and Intek conditions when reviewed by an expert interviewer, Intek deception detection accuracy increased 102% and overall accuracy 28%. We conclude that this disparity in results points to the Intek application as the major contributor to the successful overall results. However, future studies have been funded to run our conditions in different orders, which will contribute further to answering this issue.

The second limitation is the degree to which interviewee performance, nervousness, anxiety, skill as a deceiver and experience affected

the overall result. Interviewees are allocated to interviewers and assigned as a truth-teller or deceiver at random and study Section 5.1.1.1 shows interviewee experience and occupation is fairly well distributed across all conditions. Also, CV items highlighted for discussion and lies introduced were performed to strict guidelines and checking. However, as with the effect of interviewer skills, the best way to confirm these results is through performing further studies with the same randomisation of participants.

Lastly, a criticism of Intek is that it is purely an interview preparation tool or a Google search aggregator. These criticisms are true to an extent, but Intek does provide a standardised broad preparation for every item of discussion and presents this preparation focussed on the task of interviewing. Intek does provide real-time search which was used by some interviewers in our study, but real-time search was not incentivised in a job interviewing scenario, a security interviewing study with no preparation materials would test this functionality fully. Intek provides information from sources that are not easily accessible from a Google search, such as demographic information for a postcode, Intek therefore goes deeper than a standard Google search. Finally, Intek has proved to be a very effective preparation tool!

6.4 FUTURE WORK

The work presented in this thesis opened up a number of areas for potential future study. We now give an overview of this work, which may be categorised as follows: immediate Intek improvements for use in future studies; extensions of Intek for use in other interviewing scenarios; amendments to Intek for deception detection research; extensions of our web NLP work.

We discussed the possibility of future studies in Section 6.3 to address some of the other possible factors in Intek's results. Before these studies are undertaken, it would be ideal if the Intek issues and solutions identified in results Section 5.2.6 were addressed. To summarise, these issues are: a lack of useful information returned in certain circumstances, especially very generic and obscure searches; interviewees with multiple home addresses; a more obvious indication of ambiguous search results; an indication of potentially out of date information based on web site age. Additionally, we could leverage the factoid relevance data we have collected through the Intek UI to train a classifier to greatly improve the indication of useful fact-

oids on a given topic. We might also be able to auto-highlight facts of interest within these factoids using this data. This relevance data could also be used to remove factoids with low relevance, reducing UI clutter.

Looking beyond the short term, Intek might be extended to other interviewing scenarios or by incorporating aspects of professional job interviewing. The real-time search aspects of Intek might be tested in security interviewing, in which there is no material for preparation. In this scenario the relevance indication previously mentioned would be essential, as interviewers will not have time to review the whole range of factoids as in job interviewing. Intek might be extended within job interviewing by adding a number of features: integration with candidate management; automatic preparation from a CV by using NER; automatic highlighting of potential questions using a relevance classifier; add sources of information from competence and psychometrics while decreasing the amount of deception detection information, to balance the interview; visualisation of candidate alignment with company objectives. These added factors should allow Intek to deliver a fair standardised interview experience. Lastly, the interviewer might be removed entirely and Intek content delivered through online multiple choice. This approach would require auto-preparation and fact-selection in tandem to generate questions with automatic question generation to create alternative answers. It is also possible that monitoring of response characteristics, perhaps including textual statements, might be analysed in real-time for deception.

Intek is a practical interviewing tool, but it can also be used to manipulate factors of interviewer performance for the purpose of deception detection research. We identified in results Section 5.2.5 that interviewer judgement was an area that Intek does not directly support. Support for judgement might be aided by the incorporation in real-time of linguistic analysis following techniques presented in related work Section 2.2.3. Intek might also be used to manipulate test questioning in multiple ways that might affect deception detection performance: limit test questioning to only perceptual or contextual information, which may increase cognitive load on deceivers; limit test questioning to a smaller number of the best performing factoids, which may improve usability and questioning quality, but reduce interviewer familiarity with the topic; the use of expert interviewers might improve the delivery of more diagnostic questions, but they may be resistant to the CCE/Intek process.

Our work in web NER, including HTML tags in sequence-based NLP, might well be extended to other NLP tasks: relation extraction, knowledge base population, language model generation and even natural language understanding on the web. Our positive results came from a simple representation of an HTML tag as a single token in a sentence. This representation might well be expanded to incorporate information about tag position in the HTML structure or visual style information. This might lead to a joint context extraction and NER approach, in which entities are extacted only from desired areas of a web page.

7

APPENDICES

7.1 APPENDIX 1

Figure 7.1 shows the entire hierarchical task analysis for the task of CCE interviewing, as used in Chapter 3 Section 3.5.2.

Conduct CCE interview	Main tasks	Sub-tasks	Sub-sub tasks			Notes	System Interaction
Plan: Do 1. then do 2. then repeat (do 3 then 4) until required outcomes (1.6) reached, then do 5							
	1. Plan interview					Task intended to pick up on the P pf PEACE	
	Do in parallel (1.1, 1.2 , and 1.3), then if relevant do 1.4., then do in parallel (1.5 and 1.6) then do 1.7	1.1. Determine interview type				e.g., securtity screening, recruitment, suspect, witness, victim, CHIS, etc.	
		1.2.Determine interview context				e.g.,in transit, at scene, remote, delayed or real-time, threat level, time available, etc.	
		Do 1.2.1, 1.2.2 and 1.2.3 in parallel	1.2.1 Determine time available				
			1.2.2 Determine resources available				
			1.2.3 Identify interviewers			If there is more than one interviewr	
		1.3 Determine interviewee characteristics					
		Do 1.3.1 then if required to 1.3.2 and 1.3.3 in parallel	1.3.1 Identify interviewee characteristics			e.g., vulnerable, known/stranger, child	
			1.3.2 Access interview protocol			e.g., ABE	
			1.3.3 Select interview techniques			e..g, CI mnemonics, additional techniques like polygraph, etc.	
		1.4 Evaluate known information/evidence					
		Plan: If relevant do 1.4.1, then do 1.4.2 and (if relevant) 1.4.3 in parallel	1.4.1 Review information/evidence				
			1.4.2 Identify topics				
			1.4.3 Consider role of information/evidence in interview				
		1.5.Determine interview goals				e.g., detect deception, gather best evidence, gain security confidence	
		1.6. Determine required outcomes					
		1.7 Plan interview approach					
		Plan: Do 1.7.1 then 1.7.2 then 1.7.3 then 1.7.4	1.7.1 Organise facilities			e.g., if ABE then video recording	
			1.7.2 Plan topic order			e.g., to ensure all evidence/requirements are covered	
			1.7.3 Plan interview sequence			e.g., place when mnemonics will occur	
			1.7.4 Plan interview recording			e.g., if notes are taken, then what form; if multiple interviewers, who records what, etc.	
	2 Conduct baseline					Stage 1 of CCE	
		2.01 Select interviewer and interviewee					Select interviewer and interviewee
	Plan: Do 2.1 then repeat 2.2 and 2.3 in parallel until baseline judgement reached	2.1 Initiate interview					
		Plan: Do 2.1.1 then 2.1.2 then 2.1.3 then 2.1.4	2.1.1 Greet interviewee				
			2.1.2. Introduce panel				
			2.1.3 Explain interview purpose and structure				
			2.1.4 Ask interviewee if they have any needs/questions				
		2.2 Ask neutral questions				Aim is to get the interviewee to talk and to demonstrate disposition when under no challenge	
		2.3 Observe interviewee disposition					
	3 Ask information gathering Q					Stage 2 of CCE	
	Do 3.1 then 3.2 then 3.3, 3.4 and (if necessary) 3.5 in parallel	3.1 Select topic(s)				From a system perspective 3.1 can also happen during preparation (1.4)	
		Plan: Do either 3.1.1 or 3.1.2	3.1.1 Choose from presented information (e.g., CV; witness statement, evidence, baseline)				Enter desired information
			3.1.2 Choose from generic topic list			This maps onto topics to examine identity and intent; home, work, hobbies, health, education	Select topic
		3.2 Design IG question					
		Plan: Do 3.2.1 then 3.2.2 then 3.2.3	3.2.1 Choose temporality			Past, present and future to increase cognitive load	
			3.2.2 Choose question structure			Typically an IG question will be a TED Q	
			3.2.3 Plan question wording				Select IGQ from (topic-specific) list
		3.3 Ask IG question					
		3.4 Note answer					
		Plan: Do 3.4.1 then 3.4.2 then 3.4.3 then 3.4.4	3.4.1 Listen to answer				
			3.4.2 Record answer			e.g., If interview not audio/video recorded, make written notes	
			3.4.3 Clarify answer			e.g., if answer misheard or unclear	
			3.4.4 Evaluate answer				
		3.5 Ask follow-up question					
		Plan: Do 3.5.1 then 3.5.2 then 3.5.3	3.5.1 Identify follow-up topic				
			3.5.2 Choose question structure			Typically a focussed (5WH) qustion)	
			3.5.3 Repeat 3.4				
	4 Ask test Q					Stage 3 of CCE	
	Do 4.1 then if necessary 4.2 then 4.3 then 4.4 then in parallel 4.5 and 4.6	4.1 Identify expected knowledge					
		Plan: Do 4.1.1 and 4.1.2 in parallel	4.1.1 Identify experience 'range'			How long ago, for how long, how often, how important etc. is the episodic experience of the questioner	
			4.1.2Identify relevant episodic experience			i.e., determine whether knowledge requires personal experience or could be accessed generically	
		4.2 Ask bridging question					
		4.3 Choose test topic					Select topic
		4.4 Design test question					Enter desired information
		Plan: Do 4.4.1 then 4.4.2 then 4.4.3	4.4.1 Choose temporality			Past, present and future to increase cognitive load	Select TQ from (dynamic) list, recap from previous list
			4.4.2 Choose question structure			Typically an IG question will be a TED Q	
			4.4.3 Plan question wording				
		4.5 Ask test q					
		4.6 Listen to answer and observe					Feedback on TQ quality
		Plan: Do 4.6.1 and if required do 4.6.2 then 4.6.3	4.6.1 Repeat 3.4				
			4.6.2 Check answer against known information				
			4.6.3 Repeat 3.5				
	5. End interview						
		5.1. Summarise interview					
		5.2 Address interviewee questions					
		Plan: Do 5.2.1 then 5.2.2	5.2.1 Ask interviewee for questions				
			5.2.2 Provide answers to questions				
		5.3. Terminate interview					
		Plan: Do 5.3.1 then 5.3.2 then 5.3.3	5.3.1 Communicate next steps				
			5.3.2 Thank interviewee				
			5.3.3 End interview			Turn off recording devices	
	6 Evaluate interview outcomes					: Depends on nature of interview (e.g., HR interviews will typically have stages 5.2-5.4, witness interviews may not	
	Plan: Do 5.1 then if necessary do 5.2 then 5.3, then if necessary do 5.4	5.1 Document information gathered				e.g., label and store recordings; make post-interview notes, etc.	
		5.2 Discuss interview outcomes					
		5.3 Make outcome decision					
		5.4 Communicate outcome decision					

Figure 7.1 Intek hierarchical task analysis based on user goals for the CCE interviewing task.

R. Abrams. Walmart vice president forced out for lying about degree, 2014. URL https://www.nytimes.com/2014/09/17/business/17tovar.html. (Cited on page 2.)

R. Agerri and G. Rigau. Robust multilingual named entity recognition with shallow semi-supervised features. *Artificial Intelligence*, 238:63–82, 2016. (Cited on page 24.)

A. Akbik, D. Blythe, and R. Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649, 2018. (Cited on page 26.)

E. Alfonseca and S. Manandhar. An unsupervised method for general named entity recognition and automated concept discovery. In *Proceedings of the 1st international conference on general WordNet, Mysore, India*, pages 34–43, 2002. (Cited on page 29.)

R. Alfred, L. C. Leong, C. K. On, and P. Anthony. Malay named entity recognition based on rule-based approach. 2014. (Cited on page 29.)

M. Althobaiti, U. Kruschwitz, and M. Poesio. Combining minimally-supervised methods for arabic named entity recognition. *Transactions of the Association for Computational Linguistics*, 3:243–255, 2015. (Cited on page 28.)

R. K. Ando, T. Zhang, and P. Bartlett. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(11), 2005. (Cited on page 24.)

E. Apostolova and N. Tomuro. Combining visual and textual features for information extraction from online flyers. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1924–1929, 2014. (Cited on page 30.)

C. Ashby and D. Weir. Leveraging html in free text web named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 407–413, 2020. (Cited on page v.)

I. Augenstein, D. Maynard, and F. Ciravegna. Distantly supervised web relation extraction for knowledge base population. *Semantic Web*, 7(4):335–349, 2016. (Cited on page 28.)

A. Bangor, P. T. Kortum, and J. T. Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594, 2008. (Cited on page 142.)

R. Bekkerman and A. McCallum. Disambiguating web appearances of people in a social network. In *Proceedings of the 14th international conference on World Wide Web*, pages 463–470, 2005. (Cited on page 29.)

S. Bird, E. Klein, and E. Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009. (Cited on pages 29, 91, and 92.)

C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia-a crystallization point for the web of data. *Journal of web semantics*, 7(3): 154–165, 2009. (Cited on page 95.)

J. P. Blair, T. R. Levine, and A. S. Shaw. Content in context improves deception detection accuracy. *Human Communication Research*, 36(3):423–442, 2010. (Cited on page 16.)

S. Blohm. *Large-scale pattern-based information extraction from the world wide web*. KIT Scientific Publishing, 2011. (Cited on page 30.)

C. F. Bond Jr and B. M. DePaulo. Accuracy of deception judgments. *Personality and social psychology Review*, 10(3):214–234, 2006. (Cited on pages 2, 5, 10, 119, and 151.)

M. Bronzi, V. Crescenzi, P. Merialdo, and P. Papotti. Extraction and integration of partially overlapping web sources. *Proceedings of the VLDB Endowment*, 6(10):805–816, 2013. (Cited on page 94.)

J. Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996. (Cited on pages 117 and 142.)

D. B. Buller and J. K. Burgoon. Interpersonal deception theory. *Communication theory*, 6(3):203–242, 1996. (Cited on page 14.)

R. Bunescu and R. Mooney. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 576–583, 2007. (Cited on page 29.)

R. C. Bunescu. *Learning for information extraction: from named entity recognition and disambiguation to relation extraction*. PhD thesis, 2007. (Cited on pages 29 and 91.)

M. J. Cafarella, A. Y. Halevy, Y. Zhang, D. Z. Wang, and E. Wu. Uncovering the relational web. In *WebDB*, 2008. (Cited on page 90.)

C.-H. Chang and S.-C. Lui. Iepad: information extraction based on pattern discovery. In *Proceedings of the 10th international conference on World Wide Web*, pages 681–688, 2001. (Cited on pages 22 and 90.)

C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013. (Cited on page 26.)

L. Chiticariu, R. Krishnamurthy, Y. Li, F. Reiss, and S. Vaithyanathan. Domain adaptation of rule-based annotators for named-entity recognition tasks. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1002–1012, 2010. (Cited on page 24.)

L. Chiticariu, M. Danilevsky, Y. Li, F. Reiss, and H. Zhu. Systemt: Declarative text understanding for enterprise. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 76–83, 2018. (Cited on pages 23 and 24.)

J. P. Chiu and E. Nichols. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016. (Cited on pages 25 and 26.)

F. Chollet et al. keras, 2015. (Cited on page 97.)

F. Clemens, P. A. Granhag, L. A. Strömwall, A. Vrij, S. Landström, E. R. a. Hjelmsäter, and M. Hartwig. Skulking around the dinosaur: Eliciting cues to children's deception via strategic disclosure of evidence. *Applied Cognitive Psychology*, 24(7): 925–940, 2010. (Cited on page 16.)

F. Clemens, P. A. Granhag, and L. A. Strömwall. Eliciting cues to false intent: A new application of strategic interviewing. *Law and Human Behavior*, 35(6):512–522, 2011. (Cited on page 16.)

M. Collins and Y. Singer. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999. (Cited on page 29.)

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12 (ARTICLE):2493–2537, 2011. (Cited on page 25.)

V. Crescenzi, G. Mecca, P. Merialdo, et al. Roadrunner: Towards automatic data extraction from large web sites. In *VLDB*, volume 1, pages 109–118, 2001. (Cited on pages 22 and 90.)

K. Crockett, J. O'Shea, and W. Khan. Automated deception detection of males and females from non-verbal facial micro-gestures. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020. (Cited on pages 5 and 19.)

D. M. H. Cunningham and K. Bontcheva. *Text Processing with GATE (Version 6).* University of Sheffield D, 2011. (Cited on pages 23 and 24.)

B. B. Dalvi, W. W. Cohen, and J. Callan. Websets: Extracting sets of entities from the web using unsupervised information extraction. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 243–252, 2012. (Cited on pages 22 and 90.)

B. M. DePaulo, J. J. Lindsay, B. E. Malone, L. Muhlenbruck, K. Charlton, and H. Cooper. Cues to deception. *Psychological bulletin*, 129(1):74, 2003. (Cited on pages 5 and 15.)

J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018a. URL http://arxiv.org/abs/1810.04805. (Cited on pages 84 and 90.)

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018b. (Cited on pages 26 and 96.)

D. Downey, M. Broadhead, and O. Etzioni. Locating complex named entities in web text. In *IJCAI*, volume 7, pages 2733–2739, 2007a. (Cited on page 29.)

D. Downey, S. Schoenmackers, and O. Etzioni. Sparse information extraction: Unsupervised language models to the rescue. Technical report, WASHINGTON UNIV SEATTLE DEPT OF COMPUTER SCIENCE AND ENGINEERING, 2007b. (Cited on page 29.)

A. Ekbal, S. Saha, and D. Singh. Active machine learning technique for named entity recognition. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, pages 180–186, 2012. (Cited on page 91.)

P. Ekman and W. V. Friesen. Nonverbal leakage and clues to deception. *Psychiatry*, 32(1):88–106, 1969. (Cited on page 14.)

F. Enos. *Detecting deception in speech*. Citeseer, 2009. (Cited on pages 5, 18, and 21.)

O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005. (Cited on pages 28 and 30.)

O. Etzioni, A. Fader, J. Christensen, S. Soderland, et al. Open information extraction: The second generation. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011. (Cited on page 28.)

A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1535–1545, 2011. (Cited on page 28.)

T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze. Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 80–88, 2010. (Cited on page 27.)

J. R. Finkel and C. D. Manning. 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, 2009. (Cited on page 24.)

R. P. Fisher and R. E. Geiselman. *Memory enhancing techniques for investigative interviewing: The cognitive interview*. Charles C Thomas Publisher, 1992. (Cited on page 12.)

D. Freitag. Information extraction from html: Application of a general machine learning approach. In *AAAI/IAAI*, pages 517–523, 1998. (Cited on page 30.)

B. Garrison. *Computer-assisted reporting*. Routledge, 2020. (Cited on page 21.)

W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krüpl, and B. Pollak. Towards domain-independent information extraction from web tables. In *Proceedings of the 16th international conference on World Wide Web*, pages 71–80, 2007. (Cited on page 90.)

D. Gerber and A.-C. N. Ngomo. Extracting multilingual natural-language patterns for rdf predicates. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 87–96. Springer, 2012. (Cited on page 27.)

B. Gibbons. Pilot allegedly faked cv and lied about flying experience to get job with british airways, 2020. URL `https://www.birminghammail.co.uk/news/uk-news/pilot-allegedly-faked-cv-lied-19062326`. (Cited on page 1.)

D. Gillick, C. Brunk, O. Vinyals, and A. Subramanya. Multilingual language processing from bytes. *arXiv preprint arXiv:1512.00103*, 2015. (Cited on page 25.)

M. Goebel and M. Ceresna. Wrapper induction., 2009. (Cited on page 91.)

T. Gogar, O. Hubacek, and J. Sedivy. Deep neural networks for web page information extraction. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 154–163. Springer, 2016. (Cited on pages 23, 30, and 90.)

T. Green, B. Ord, and G. Shaw. *Investigative interviewing explained*. LexisNexis Butterworths, 2008. (Cited on page 13.)

Q. Hao, R. Cai, Y. Pang, and L. Zhang. From one tree to a forest: a unified solution for structured web data extraction. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 775–784, 2011. (Cited on page 94.)

M. Hartwig, P. A. Granhag, L. A. Strömwall, and A. Vrij. Detecting deception via strategic disclosure of evidence. *Law and human behavior*, 29(4):469–484, 2005. (Cited on page 16.)

M. Hartwig, P. A. Granhag, L. A. Strömwall, and O. Kronkvist. Strategic use of evidence during police interviews: When training to detect deception works. *Law and human behavior*, 30(5):603, 2006. (Cited on page 16.)

M. Hartwig, P. A. Granhag, L. Stromwall, A. G. Wolf, A. Vrij, and E. R. a. Hjelmsäter. Detecting deception in suspects: Verbal cues as a function of interview strategy. *Psychology, Crime & Law*, 17(7):643–656, 2011. (Cited on page 16.)

N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A. K. Nayak, et al. Claimbuster: The first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment*, 10(12):1945–1948, 2017. (Cited on page 19.)

C. A. Henle, B. R. Dineen, and M. K. Duffy. Assessing intentional resume deception: Development and nomological network of a resume fraud measure. *Journal of Business and Psychology*, 34(1):87–106, 2019. (Cited on pages 1 and 2.)

M. Honnibal and I. Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017. (Cited on page 83.)

Z. Huang, W. Xu, and K. Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015. (Cited on page 25.)

F. E. Inbau, J. E. Reid, J. P. Buckley, B. C. Jayne, et al. *Essentials of the Reid technique: Criminal interrogation and confessions*. Jones & Bartlett Publishers, 2013. (Cited on page 13.)

K. Inoue, K. Hara, D. Lala, S. Nakamura, K. Takanashi, and T. Kawahara. A job interview dialogue system with autonomous android erica. In *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction: 10th International Workshop on Spoken Dialogue Systems*, pages 291–297. Springer Singapore, 2021. (Cited on page 20.)

J. Jiang and C. Zhai. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 264–271, 2007. (Cited on page 27.)

P. Jiménez and R. Corchuelo. On learning web information extraction rules with tango. *Information Systems*, 62:74–103, 2016. (Cited on page 23.)

M. K. Johnson and C. L. Raye. Reality monitoring. *Psychological review*, 88(1):67, 1981. (Cited on page 17.)

M. M. H. Kim. Incremental knowledge acquisition approach for information extraction on both semi-structured and unstructured text from the open domain web. In *Proceedings of the Australasian Language Technology Association Workshop 2017*, pages 88–96, 2017. (Cited on page 30.)

T. Kiss and J. Strunk. Unsupervised multilingual sentence boundary detection. *Computational linguistics*, 32(4):485–525, 2006. (Cited on page 29.)

C. Kohlschütter, P. Fankhauser, and W. Nejdl. Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 441–450, 2010. (Cited on page 30.)

G. Köhnken. Statement validity analysis and the'detection of the truth'in p.-a. granhag & l. strömwall (eds.), the detection of deception in forensic contexts (p. 41–63), 2004. (Cited on page 17.)

G. Köhnken, R. Milne, A. Memon, and R. Bull. The cognitive interview: A meta-analysis. *Psychology, Crime and Law*, 5(1-2):3–27, 1999. (Cited on page 13.)

F. Kokkoras, K. Ntonas, and N. Bassiliades. Deixto: A web data extraction suite. In *Proceedings of the 6th Balkan Conference in Informatics*, pages 9–12, 2013. (Cited on page 23.)

Z. Kozareva. Bootstrapping named entity recognition with automatically generated gazetteer lists. In *Student Research Workshop*, 2006. (Cited on page 27.)

G. Krupka and K. Hausman. Isoquest inc.: description of the netowl extractor system as used for muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*, 1998. (Cited on page 24.)

J. Levashina and M. A. Campion. Measuring faking in the employment interview: development and validation of an interview faking behavior scale. *Journal of applied psychology*, 92(6):1638, 2007. (Cited on page 1.)

T. R. Levine. A few transparent liars explaining 54% accuracy in deception detection experiments. *Annals of the International Communication Association*, 34(1):41–61, 2010. (Cited on page 3.)

T. R. Levine. New and improved accuracy findings in deception detection research. *Current Opinion in Psychology*, 6:1–5, 2015. (Cited on pages 3 and 15.)

T. R. Levine and S. A. McCornack. Behavioral adaptation, confidence, and heuristic-based explanations of the probing effect. *Human Communication Research*, 27(4): 471–502, 2001. (Cited on page 16.)

T. R. Levine, R. K. Kim, and J. P. Blair. (in) accuracy at detecting true and false confessions and denials: An initial test of a projected motive model of veracity judgments. *Human Communication Research*, 36(1):82–102, 2010. (Cited on pages 2 and 4.)

T. R. Levine, J. P. Blair, and D. D. Clare. Diagnostic utility: Experimental demonstrations and replications of powerful question effects in high-stakes deception detection. *Human Communication Research*, 40(2):262–289, 2014a. (Cited on page 16.)

T. R. Levine, D. D. Clare, J. P. Blair, S. McCornack, K. Morrison, and H. S. Park. Expertise in deception detection involves actively prompting diagnostic information rather than passive behavioral observation. *Human Communication Research*, 40(4): 442–462, 2014b. (Cited on page 16.)

S. I. Levitan, A. Maredia, and J. Hirschberg. Linguistic cues to deception and perceived deception in interview dialogues. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1941–1950, 2018. (Cited on pages 5 and 19.)

M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019. (Cited on page 87.)

LexisNexis. Legal and professional solutions and products, Jun 2021. URL https://www.lexisnexis.co.uk/. (Cited on page 21.)

Y. Li, K. Bontcheva, and H. Cunningham. Using uneven margins svm and perceptron for information extraction. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 72–79, 2005. (Cited on page 24.)

Z. Li and W. K. Ng. Wiccap: From semi-structured data to structured data. In *Proceedings. 11th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, 2004.*, pages 86–93. IEEE, 2004. (Cited on page 22.)

R. Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932. (Cited on page 116.)

Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, 2016. (Cited on page 28.)

X. Ma and E. Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*, 2016. (Cited on pages 25, 90, and 96.)

J. Masip, S. L. Sporer, E. Garrido, and C. Herrero. The detection of deception with the reality monitoring approach: A review of the empirical evidence. *Psychology, Crime & Law*, 11(1):99–122, 2005. (Cited on page 17.)

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. (Cited on pages 24 and 96.)

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. 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 1003–1011, 2009. (Cited on pages 27, 28, 93, and 95.)

M. M. Mirończuk. The biggrams: the semi-supervised information extraction system from html: an improvement in the wrapper induction. *Knowledge and Information Systems*, 54(3):711–776, 2018. (Cited on pages 23, 30, and 91.)

R. Molich and J. Nielsen. Improving a human-computer dialogue. *Communications of the ACM*, 33(3):338–348, 1990. (Cited on page 51.)

D. Nadeau, P. D. Turney, and S. Matwin. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Conference of the Canadian society for computational studies of intelligence*, pages 266–277. Springer, 2006. (Cited on page 29.)

G. Nahari, A. Vrij, and R. P. Fisher. Does the truth come out in the writing? scan as a lie detection tool. *Law and Human Behavior*, pages 1–11, 2011. (Cited on page 18.)

G. Nahari, A. Vrij, and R. P. Fisher. Exploiting liars' verbal strategies by examining the verifiability of details. *Legal and Criminological Psychology*, 19(2):227–239, 2014a. (Cited on page 18.)

G. Nahari, A. Vrij, and R. P. Fisher. The verifiability approach: Countermeasures facilitate its ability to discriminate between truths and lies. *Applied Cognitive Psychology*, 28(1):122–128, 2014b. (Cited on page 18.)

C. Nédellec, R. Bossy, J.-D. Kim, J.-J. Kim, T. Ohta, S. Pyysalo, and P. Zweigenbaum. Overview of bionlp shared task 2013. In *Proceedings of the BioNLP shared task 2013 workshop*, pages 1–7, 2013. (Cited on page 91.)

F. Niu, C. Zhang, C. Ré, and J. W. Shavlik. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. *VLDS*, 12:25–28, 2012. (Cited on page 23.)

J. Nothman, N. Ringland, W. Radford, T. Murphy, and J. R. Curran. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151–175, 2013. (Cited on page 28.)

T. C. Ormerod. Using task analysis as a primary design method: The sgt approach. *Cognitive task analysis*, pages 181–200, 2000. (Cited on page 50.)

T. C. Ormerod and C. J. Dando. Finding a needle in a haystack: Toward a psychologically informed method for aviation security screening. *Journal of Experimental Psychology: General*, 144(1):76, 2015. (Cited on pages 3, 5, 6, 43, and 109.)

T. C. Ormerod and A. Shepherd. Using task analysis for information requirements specification: The sub-goal template (sgt) method. *The handbook of task analysis for human-computer interaction*, page 347, 2003. (Cited on page 50.)

J. O'Shea, K. Crockett, W. Khan, P. Kindynis, A. Antoniades, and G. Boultadakis. Intelligent deception detection through machine based interviewing. In *2018 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2018. (Cited on pages 5 and 19.)

H. S. Park, T. Levine, S. McCornack, K. Morrison, and M. Ferrara. How people really detect lies. *Communication Monographs*, 69(2):144–157, 2002. (Cited on page 15.)

A. Passos, V. Kumar, and A. McCallum. Lexicon infused phrase embeddings for named entity resolution. *arXiv preprint arXiv:1404.5367*, 2014. (Cited on page 24.)

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162. (Cited on page 96.)

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pages 2227–2237, 2018. URL www.scopus.com. Cited By :2761. (Cited on page 25.)

T. Phillips, R. K. Saunders, J. Cossman, and E. Heitman. Assessing trustworthiness in research: a pilot study on cv verification. *Journal of Empirical Research on Human Research Ethics*, 14(4):353–364, 2019. (Cited on page 1.)

D. Putthividhya and J. Hu. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, 2011. (Cited on page 27.)

D. Qiu, L. Barbosa, X. L. Dong, Y. Shen, and D. Srivastava. Dexter: large-scale discovery and extraction of product specifications on the web. *Proceedings of the VLDB Endowment*, 8(13):2194–2205, 2015. (Cited on page 22.)

L. Qiujun. Extraction of news content for text mining based on edit distance. *Journal of Computational Information Systems*, 6(11):3761–3777, 2010. (Cited on pages 22 and 90.)

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training, 2018. (Cited on page 25.)

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. (Cited on page 26.)

H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 2931–2937, 2017. (Cited on page 19.)

S. R. Reddick. Point: The case for profiling. *International Social Science Review*, 79 (3/4):154–156, 2004. (Cited on page 15.)

R. Rehurek and P. Sojka. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer, 2010. (Cited on page 97.)

M.-A. Reinhard, S. L. Sporer, M. Scharmach, and T. Marksteiner. Listening, not watching: Situational familiarity and the ability to detect deception. *Journal of Personality and Social Psychology*, 101(3):467, 2011. (Cited on page 16.)

M.-A. Reinhard, S. L. Sporer, and M. Scharmach. Perceived familiarity with a judgmental situation improves lie detection ability. *Swiss Journal of Psychology*, 2012. (Cited on page 16.)

M.-A. Reinhard, M. Scharmach, and P. Müller. It's not what you are, it's what you know: Experience, beliefs, and the detection of deception in employment interviews. *Journal of Applied Social Psychology*, 43(3):467–479, 2013. (Cited on page 3.)

Revision.ai. Award winning ai-quiz generator, Jun 2021. URL https://www.revision.ai/. (Cited on page 20.)

J. Richardson, T. C. Ormerod, and A. Shepherd. The role of task analysis in capturing requirements for interface design. *Interacting with computers*, 9(4):367–384, 1998. (Cited on page 50.)

L. Richardson. Beautiful soup documentation. *Dosegljivo: https://www. crummy. com/software/BeautifulSoup/bs4/doc/.[Dostopano: 7. 7. 2018]*, 2007. (Cited on pages 29 and 91.)

A. Ritter, S. Clark, O. Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1524–1534, 2011. (Cited on page 28.)

B. Roth, T. Barth, M. Wiegand, and D. Klakow. A survey of noise reduction methods for distant supervision. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 73–78, 2013. (Cited on page 95.)

N. Roulin, A. Bangerter, and J. Levashina. Interviewers' perceptions of impression management in employment interviews. *Journal of Managerial Psychology*, 2014. (Cited on page 2.)

M. A. Russell. *Mining the social web: data mining Facebook, Twitter, LinkedIn, Google+, GitHub, and more*. " O'Reilly Media, Inc.", 2013. (Cited on page 91.)

E. F. Sang and F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003. (Cited on pages 91, 96, and 97.)

A. Sapir. The lsi course on scientific content analysis (scan). *Phoenix, ZA: Laboratory for Scientific Interrogation*, 1987. (Cited on page 18.)

P. R. SB, M. Agnihotri, and D. B. Jayagopi. Automatic follow-up question generation for asynchronous interviews. In *Proceedings of the Workshop on Intelligent Information Processing and Natural Language Generation*, pages 10–20, 2020. (Cited on page 20.)

T. D. Science and U. Technology Laboratory. Relationship and entity extraction evaluation dataset, 2017. URL `https://github.com/dstl/re3d`. (Cited on page 93.)

SELECTPro. Selection interview guides, Jun 2021. URL `http://www.selectpro.net/`. (Cited on page 20.)

D. Shen, J. Zhang, J. Su, G. Zhou, and C. L. Tan. Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 589–596, 2004. (Cited on page 27.)

Y. Shen, H. Yun, Z. C. Lipton, Y. Kronrod, and A. Anandkumar. Deep active learning for named entity recognition. *arXiv preprint arXiv:1707.05928*, 2017. (Cited on page 27.)

E. W. Shepard. Resistance in interviews: The contribution of police perceptions and behaviour. *Issues in Criminological & Legal Psychology*, 1993. (Cited on page 12.)

E. Shepherd. Developing interviewing skills: A career span perspective. *New directions in police training. London: HMSO*, 1988. (Cited on page 12.)

U. Singh, V. Goyal, and G. S. Lehal. Named entity recognition system for urdu. In *Proceedings of COLING 2012*, pages 2507–2518, 2012. (Cited on page 29.)

H. A. Sleiman and R. Corchuelo. A class of neural-network-based transducers for web information extraction. *Neurocomputing*, 135:61–68, 2014. (Cited on page 23.)

S. G. Small and L. Medsker. Review of information extraction technologies and applications. *Neural computing and applications*, 25(3-4):533–548, 2014. (Cited on page 91.)

S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine learning*, 34(1-3):233–272, 1999. (Cited on pages 30 and 91.)

R. Speck and A.-C. N. Ngomo. Ensemble learning for named entity recognition. In *International semantic web conference*, pages 519–534. Springer, 2014. (Cited on page 91.)

S. L. Sporer. Reality monitoring and detection of deception. *The detection of deception in forensic contexts*, pages 64–102, 2004. (Cited on page 17.)

L. Sterckx, T. Demeester, J. Deleu, and C. Develder. Using active learning and semantic clustering for noise reduction in distant supervision. In *4e Workshop on Automated Base Construction at NIPS2014 (AKBC-2014)*, pages 1–6, 2014. (Cited on page 28.)

M.-H. Su, C.-H. Wu, K.-Y. Huang, Q.-B. Hong, and H.-H. Huang. Follow-up question generation using pattern-based seq2seq with a small corpus for interview coaching. In *INTERSPEECH*, pages 1006–1010, 2018. (Cited on page 20.)

M.-H. Su, C.-H. Wu, and Y. Chang. Follow-up question generation using neural tensor network-based domain ontology population in an interview coaching system. In *INTERSPEECH*, pages 4185–4189, 2019. (Cited on page 20.)

M. Surdeanu, D. McClosky, J. Tibshirani, J. Bauer, A. X. Chang, V. I. Spitkovsky, and C. D. Manning. A simple distant supervision approach for the tac-kbp slot filling task. 2010. (Cited on page 28.)

M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465, 2012. (Cited on page 28.)

D. Sweeney. *Deception Detection in Recruitment Interviewing*. PhD thesis, University of Sussex, 2022. (Cited on pages v, 103, and 119.)

TestGorilla. Pre-employment screening tests and assessments, Jun 2021. URL https://www.testgorilla.com/. (Cited on page 20.)

J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018. (Cited on page 19.)

E. F. Tjong Kim Sang and F. De Meulder. 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, 2003. URL https://www.aclweb.org/anthology/W03-0419. (Cited on page 84.)

Y. Tsunomori, G. Neubig, S. Sakti, T. Toda, and S. Nakamura. An analysis towards dialogue-based deception detection. In *Natural Language Dialog Systems and Intelligent Assistants*, pages 177–187. Springer, 2015. (Cited on page 21.)

N. C. Tu, T. T. Oanh, P. X. Hieu, and H. Q. Thuy. Named entity recognition in vietnamese free-text and web documents using conditional random fields. In *The 8th Conference on Some selection problems of Information Technology and Telecommunication*, page 12. Citeseer, 2005. (Cited on page 91.)

N. W. Twyman, S. J. Pentland, and L. Spitzley. Deception detection in online automated job interviews. In *International Conference on HCI in Business, Government, and Organizations*, pages 206–216. Springer, 2018. (Cited on page 5.)

U. S. G. A. O. USGAO. Aviation security: Tsa is taking steps to validate the science underlying its passenger behavior detection program, but efforts may not be comprehensive. 2011. (Cited on page 1.)

A. Vrij and P. A. Granhag. Eliciting cues to deception and truth: What matters are the questions asked. *Journal of Applied Research in Memory and Cognition*, 1(2):110–117, 2012. (Cited on pages 5, 15, and 20.)

W. Y. Wang. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*, 2017. (Cited on page 19.)

R. Weischedel, M. Palmer, M. Marcus, E. Hovy, S. Pradhan, L. Ramshaw, N. Xue, A. Taylor, J. Kaufman, M. Franchini, et al. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23, 2013. (Cited on page 91.)

C. Whitelaw, A. Kehlenbeck, N. Petrovic, and L. Ungar. Web-scale named entity recognition. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 123–132, 2008. (Cited on pages 27 and 91.)

A. S. Wibawa and A. Purwarianti. Indonesian named-entity recognition for 15 classes using ensemble supervised learning. *Procedia Computer Science*, 81:221–228, 2016. (Cited on page 91.)

T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019. (Cited on page 97.)

E. Wong. User interface design guidelines: 10 rules of thumb, 2020. URL https://www.interaction-design.org/literature/article/user-interface-design-guidelines-10-rules-of-thumb. (Cited on page 51.)

D. Wu, W. S. Lee, N. Ye, and H. L. Chieu. Domain adaptive bootstrapping for named entity recognition. In *EMNLP'09 Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Volume 3*, pages 1523–1532. Association for Computing Machinery, 2009. (Cited on page 27.)

V. Yadav, R. Sharp, and S. Bethard. Deep affix features improve neural named entity recognizers. In *Proceedings of the seventh joint conference on lexical and computational semantics*, pages 167–172, 2018. (Cited on page 25.)

A. Yates, M. Banko, M. Broadhead, M. J. Cafarella, O. Etzioni, and S. Soderland. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, 2007. (Cited on page 28.)

J. Zeng and Y. I. Nakano. Exploiting a large-scale knowledge graph for question generation in food preference interview systems. In *Proceedings of the 25th International Conference on Intelligent User Interfaces Companion*, pages 53–54, 2020. (Cited on page 20.)

Y. Zhai and B. Liu. Structured data extraction from the web based on partial tree alignment. *IEEE Transactions on Knowledge and Data Engineering*, 18(12):1614–1628, 2006. (Cited on page 22.)

S. Zhang and N. Elhadad. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of biomedical informatics*, 46(6): 1088–1098, 2013. (Cited on page 29.)

M. Zuckerman, B. M. DePaulo, and R. Rosenthal. Verbal and nonverbal communication of deception. In *Advances in experimental social psychology*, volume 14, pages 1–59. Elsevier, 1981. (Cited on page 14.)

Zyte. Best practices for web scraping [accessed september 8, 2021], 2020. URL https://www.zyte.com/learn/web-scraping-best-practices/. (Cited on page 75.)