Extracting and Attributing Quotes in Text and Assessing them as Opinions

Tim O'Keefe

Supervisors: Dr. Irena Koprinska and Dr. James Curran

THE UNIVERSITY OF
SYDNEY

A thesis submitted
in fulfilment of the requirements
for the degree of Doctor of Philosophy

School of Information Technologies
Faculty of Engineering & IT
The University of Sydney

2014

Abstract

News articles often report on the opinions that salient people have about important issues. While it is possible to infer an opinion from a person's actions, it is much more common to demonstrate that a person holds an opinion by reporting on what they have said. These instances of speech are called either quotes or reported speech, and in this thesis we set out to detect instances of reported speech, attribute them to their speaker, and to identify which instances provide evidence of an opinion.

We first focus on extracting reported speech. This problem involves finding all acts of communication that are reported within an article, specifically: direct quotes, which are literally transcribed and surrounded by quotation marks; indirect quotes, which are not literally transcribed and are not marked orthographically; and mixed quotes, which are part direct and part indirect. Previous work has generally approached this task with rule-based methods, however there are several factors that confound these approaches. To demonstrate this, we build a corpus of 965 news articles, where we mark all instances of direct, indirect, and mixed speech. We then show that a supervised token-based approach outperforms all of our rule-based alternatives, even in extracting direct quotes. This thesis represents the first large-scale evaluation of this task.

Next, we examine the problem of finding the speaker of each quote. For this task we annotate the same 965 news articles with links from each quote to its speaker. Using this, and three other corpora, we develop new methods and features for quote attribution, which achieve state-of-the-art accuracy on our corpus and strong results on the others. We also provide a thorough examination of what it means to be the speaker of a quote. This leads us to examine and evaluate the effects of coreference resolution on quote attribution, and as part of this we show that coreference resolution is too inaccurate to be used in literature, though it performs reasonably well for news. We end our work on quote attribution by evaluating the full pipeline of systems, and by demonstrating that we can use named entity linking to link quotes to a knowledge base.

Having extracted quotes and determined who spoke them, we move on to the opinion mining part of our work. Most of the task definitions in opinion mining do not easily work with opinions in news, particularly complex opinions about topics of debate. We therefore define a new task, where the aim is to classify whether quotes demonstrate support, neutrality, or opposition to a given position statement, which states clearly and unambiguously a particular opinion on a topic. We found that this formulation improved annotator agreement when compared to our earlier annotation schemes, which had aimed to mark opinions more directly. Using this we build an opinion corpus of 700 news documents covering 7 topics. In this thesis we do not attempt this full task, but we do present preliminary results on the task of identifying relevant non-neutral quotes.

This work has examined the problems of extracting speech, attributing it to its speaker, and classifying the opinions that the speech holds. For reported speech extraction and attribution we have presented a new corpus, baselines, and improved methods. For the opinion task we have presented a second corpus, analysis, and preliminary results on the task. The output of these systems is robust enough that they can be immediately used for various applications.

Acknowledgements

Even though it is generally a positive experience, getting through a thesis is at times extremely challenging. There are periods where nearly every PhD candidate ends up in the unusual position of simultaneously being extremely busy and extremely poor, and I was no exception. It's in those periods, and, to be fair, all the others, where the support of colleagues, family, and friends is absolutely essential.

I'd like to start by thanking my two supervisors, Dr. Irena Koprinska and Dr. James R. Curran. Irena was the first person to introduce me to the idea that computers could do anything meaningful with natural language, and more importantly, that *I* could make them do it. Irena's help and guidance in taking my first steps in this field were essential to the success of my PhD. Irena also deserves my thanks for pulling me aside and actually making me do a PhD in the first place. Without that intervention it's unlikely I would have started down this path, and without Irena's help in filling out all of the forms that day (which happened to be the due date), I certainly wouldn't have been accepted that year.

James' role in my PhD was no less important. As my PhD progressed the work moved more and more into James' many areas of expertise, and without James' knowledge and hard work I'd have struggled to complete even half of the work I did. I owe James particular thanks for setting up both the Computable News project and Schwa lab. While I didn't always appreciate the extra work that came with the Computable News project, it was definitely a positive part of my PhD.

The colleagues that I had during my PhD were the best anyone could hope for. Schwa lab was and remains an exceptional institution filled with truly brilliant people. I'd like to thank my cohort, namely Will Radford, Richard Billingsley, Tim Dawborn, and James

Constable. I'd also like to particularly thank Matt Honnibal, who was always happy to talk about the frustrations and to celebrate the successes that I experienced during my PhD. More broadly, I'd like to thank everyone in Schwa lab for their feedback on my papers and ultimately on the thesis itself. This document is the better for it.

While they may not have provided input on my work, my friends and family still played a crucial role in this thesis. My friends were extremely understanding when either cost or time prevented me from joining them on excursions, and this extended even to those occasions when I disappeared into the labs for weeks on end. I'd like to thank my father, Brendan, for his support and for making time to visit me when I didn't have the time to go visit him. I'd like to thank my mother, Sandra, whose prior experience in completing a PhD gave her great insight in what to ask and what not to ask. I'd like to thank my sister, Anne, for the good humour and positivity that she showed throughout this long process. Finally, I'd like to thank my closest friend, Sean, who was completing his thesis at the same time as me. I think we both benefited from having someone we know so well, who was at the same stage in the process, albeit in different fields.

Statement of compliance

I certify that:

- I have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure;
- I understand that failure to comply with the Student Plagiarism: Coursework Policy and Procedure can lead to the University commencing proceedings against me for potential student misconduct under Chapter 8 of the University of Sydney By-Law 1999 (as amended);
- this Work is substantially my own, and to the extent that any part of this Work is not my own I have indicated that it is not my own by Acknowledging the Source of that part or those parts of the Work.

Name: *Tim O'Keefe*

Signature: Date: *11th November 2014*

Contents

List of Tables

List of Figures

List of Acronyms

ACL Association for Computational Linguistics.

AFP Agence France Presse.

AR Attribution Relations.

BBN BBN Technologies.

CFG context-free grammar.

CRF Conditional Random Field.

ETL Entropy-guided Transformation Learning.

HMM Hidden Markov Model.

IAC Internet Argument Corpus.

IOB Inside-Outside-Begin.

KB Knowledge Base.

LDA Latent Dirichlet Allocation.

LIT Literature Corpus.

ME Maximum Entropy.

MEMM Maximum Entropy Markov Model.

ML Machine Learning.

MPQA Multi-Perspective Question Answering.

NB Naïve Bayes.

NE Named Entity.

NEL Named Entity Linking.

NER Named Entity Recognition.

NLP Natural Language Processing.

NP Noun Phrases.

OPQA Opinion Question Answering.

PARC Penn Discourse Treebank Attribution Relations Corpus.

PCFG Probabilistic Context-Free Grammar.

PDTB Penn Discource TreeBank.

PLSA Probabilistic Latent Semantic Analysis.

PMI Pointwise Mutual Information.

POS Part-of-Speech.

PTB Penn TreeBank.

QA Question Answering.

RNN Recursive Neural Networks.

RNTN Recursive Neural Tensor Networks.

SMH Sydney Morning Herald.

SMHC Sydney Morning Herald Corpus.

SO-CAL Semantic Orientation CALculator.

SVM Support Vector Machine.

TAC Text Analysis Conference.

VP Verb Phrase.

WSJ Wall Street Journal.

1 · Introduction

News text often reports on important topics and events, and as such it is a valuable source from which to extract opinions. However, as a genre, news has the peculiarity that – with the exception of editorials – it is best practice for journalists to cover a topic in a neutral and unbiased fashion. This means that journalists should generally avoid inserting their own opinions into their writing, and should instead report on the opinions of salient entities. Though it is possible to demonstrate the opinions people hold by describing their actions, it is much more common to demonstrate opinions by reporting on what people say.

When text describes an act of communication, it is called reported speech. Developing computational approaches that can extract reported speech and attribute it to its speaker is useful for understanding discourse in and of itself. However, for understanding the opinions in news it plays a crucial role, as the most opinionated parts of news text are the instances of reported speech. It is therefore a useful focus for research into both opinion mining and discourse more generally.

Focusing on reported speech has other advantages. Chapters 2 and 3 will demonstrate that reported speech can be accurately extracted and attributed to the correct speaker. More importantly, reported speech has been hand-picked by journalists from statements made by a speaker. If the journalist is using this speech to demonstrate an opinion, then it will typically be on-point and concise.

In this thesis we investigate three core tasks. The first task is reported speech extraction, where the goal is to identify the spans of reported speech. Next, we examine reported speech attribution, where we aim to find who said each instance of speech. Finally, we categorise the opinions present according to a novel opinion mining scheme. In this thesis we do not attempt the full opinion task that we have defined, but we do present preliminary results in identifying opinionated speech. In the next three subsections we will describe these tasks in more detail.

Reported Speech Extraction

Reported speech[1] extraction is the task of automatically extracting instances of text where the author describes the content of an act of communication. The communicative act is not limited to speaking, and so may include writing and other forms of communication. The inclusion of the content of what was communicated is critical, as we are not interested in cases where the author notes that speech has occurred, without actually including what was said. Reported speech can be broken down into three main categories, which we illustrate with the following examples:

(1) a. "For 10 million, you can move $100 million of stocks," a specialist on the Big Board gripes.

b. Mr Walsh said Rio was continuing to hold discussions with its customers to arrive at a mutually agreed price.

c. Police would only apply for the restrictions when "we have a lot of evidence that late-night noise, anti-social behaviour and alcohol-related crime is disturbing the residents of that neighbourhood", Superintendent Tony Cooke said.

Example 1a is an instance of a *direct quote*. Direct quotes are marked orthographically through the use of quotation marks, and conventionally indicate that the journalist has transcribed what the speaker said verbatim. In some cases a verbatim transcription of an act of communication will not be possible, so in those circumstances it is common for journalists to use *indirect speech*, as in Example 1b. In instances of indirect speech the content has been reworded for brevity or clarity, while the communicative intent of the speaker is kept intact. Unlike direct quotes, indirect speech is not marked orthographically. The final example (1c) shows a mixed quote, which is contained partly within quotation marks and partly outside the marks. This indicates that the direct part is transcribed verbatim, while the indirect part has been reworded. Mixed quotes are used in cases where the journalist

[1] See Bergler (1992) for a comprehensive description of the role of reported speech in news articles.

needs to make grammatical adjustments to a direct quote, or where part of an indirect quote is transcribed verbatim, so as to distance the journalist from a particular claim or wording. We consider the task of reported speech extraction to be to extract all direct, indirect, and mixed quotes.

It is worth noting here that while our work labels all of these categories as reported speech, some authors consider only indirect speech, as in Example 1b, to be reported speech. Other authors refer to all of these categories as quotations or quotes, while still others reserve quote and quotation for direct quotes only (Example 1a). In the interests of simple and consistent terminology, we consider quote, quotation, and reported speech to be synonyms, and use these terms interchangeably. When we need to refer to one of the categories more precisely we will explicitly add the specifier, i.e. direct, indirect, or mixed.

In terms of difficulty it may appear that extracting direct quotes is relatively straightforward. This is partly true, as it is easy to extract everything that appears between quotation marks. However, note that this strategy will also extract all of the directly-quoted portions of mixed quotes. Furthermore many writers – and journalists in particular – use *scare quotes* to highlight or distance themselves from a particular wording or phrasing. This involves putting quotation marks around the word or phrase, which conventionally disassociates the writer from the phrase. This can make simple rule-based extraction of direct quotes surprisingly low precision, as not all content between quotation marks is a direct quote.

Mixed and indirect quotes present an even greater problem, as they are not marked orthographically. As such, for indirect quotes we need syntax-aware approaches that can extract the correct span of text, without the straightforward orthographic prompting that accompanies direct quotes. In this thesis we evaluate several approaches to extracting all direct, indirect, and mixed quotes.

Reported Speech Attribution

Extracting reported speech from text is not sufficient for our main goal of opinion mining. We need to go further and actually determine who spoke each instance of reported speech, so that we can attribute opinions to them. This task is called reported speech attribution. We will illustrate the task, as well as some of its difficulties with the following examples:

(2) a. "It doesn't seem the numbers are there yet, but I will continue to build my case," Senator Xenophon said.

b. Teddy Riley, who co-produced Jackson's 1991 album Dangerous, described Jackson as "the greatest". "Every day I spent with him I had fun, but. . . working on the record was a lot of hard work, because he settles for nothing less than great."

c. "The Australian Government makes no apology whatsoever for deploying the most hardline measures necessary to deal with the problems of illegal immigration into Australia," he thundered this week.

Example 2a shows the simplest case for quote attribution, where there is a speech verb (said), the speech itself, and a speaker. Extracting the speaker for this example involves finding the subject of the speech verb. However, this approach quickly becomes insufficient, as Example 2b shows. The problem in Example 2b is that the speaker is mentioned in the sentence preceding the quote, and is in no syntactic relationship with the speech itself. This example makes clear that a level of discourse-awareness is essential to systems performing this task.

Example 2c has the same form as Example 2a, but uses a pronoun instead of a proper noun. This raises another significant issue. A system that can extract the pronoun has in one sense found the speaker, however the pronoun on its own may not be very informative. In fact, most applications of quote attribution, including opinion mining, require the pronoun to be disambiguated back to a more definite mention of the speaker. This means that quote attribution systems may need to perform some level of coreference resolution, and may even need to go further to find a cross-document representation of each speaker.

Opinion Mining

Broadly speaking, opinion mining, otherwise known as sentiment analysis, is the task of categorising opinions in text according to some labelling or scoring scheme. While this definition may appear straightforward, there is considerable variance between what the proposed approaches consider to be an opinion, and in how those opinions are categorised. In its simplest form opinion mining labels a unit of text – be it a whole document, a sentence, or even a single word – as either positive or negative. The underlying assumption behind this approach is that the communicative intent behind the text is to evaluate some known topic or entity as good or bad. This view of the task is generally appropriate for texts that were written to evaluate an entity, with the classic example being product and movie reviews.

Other works have expanded on this definition by treating an opinion as a triple of an opinion holder, an opinion target, and the actual polarity of the opinion itself. For these approaches the target is usually textually anchored, and needs to be found by the system. The system also needs to explicitly assign a holder to the opinion, which can be either an entity that appears in the text, or the author of the text itself. This definition is not necessarily incompatible with the approaches that seek to identify the sentiment alone, as in those cases we can regard the text's author as the opinion holder and the item being reviewed as the opinion target. In this work we are interested in opinions that do not neatly fit into these definitions. Consider the following examples:

(3) a. "Whether it's a stealth tax, the emissions trading scheme, whether it's an upfront. . . tax like a carbon tax, there will not be any new taxes as part of the Coalition's policy"

b. "I now believe that the time has come. . . for us to have a truly Australian constitutional head of state."

c. "I am pro-life and also pro-choice and I don't find any conflict in that."

How are we to categorise these opinions? It is clear that classifying any of them as simply positive or negative gives us no information about the viewpoint that the speaker is arguing for. If we were to make such a classification it would likely depend very much upon our own point of view about the topics or events under discussion. This would make agreement about these labels low, as the relationship between the labels and the sides of the debate is ambiguous.

Adding targets for the opinions gets us a little bit further. In Example 3a there are two targets, the emissions trading scheme and a carbon tax, which the speaker is clearly negative towards. However it is worth noting that the overall statement is really a statement of fact, with the value judgement about new taxes being implicit. This example is thus somewhat difficult for the opinion schemes noted above. Example 3b similarly has a clear opinion, which is that Australia should become a republic. The problem is that there is no mention of the republic, or the opposing constitutional model, which is the constitutional monarchy. As such it is difficult to reconcile the target that is present, i.e. a truly Australian constitutional head of state, with the actual topic we are interested in, which is the question of whether Australia should become a republic. Example 3c is difficult in a different way, as the speaker is being positive towards two targets, i.e. pro-life and pro-choice, which we would usually consider to be mutually-exclusive. A scheme that classifies both of these targets as positive would miss the overall point that the speaker is trying to make.

These examples make three points clear. Firstly, evaluating these opinions with respect to ones own views will result in disagreement about whether the statements are positive or negative. Secondly, textually-grounding the target of each opinion will in many cases miss the overall viewpoint that the speaker is arguing for. Thirdly, speakers will often cite facts to reinforce their arguments, and these factual statements can still provide evidence of an opinion.

Due to these issues we propose a new scheme that better enables us to categorise the opinions in these examples. This scheme involves labelling whether each opinion indicates support for or opposition to a given *position statement*. A position statement is a clear and unambiguous statement of a particular point of view on a topic. By evaluating a given span of text against a position statement, we are able to remove much of the ambiguity coming from labels like positive and negative, as the scheme now labels statements as providing evidence of support or opposition to the given viewpoint. It also removes the problems raised by having textually-grounded targets, as the labelling is done explicitly against the issue we are interested in.

While the use of position statements clarifies how we label opinions on complex topics, they do present some drawbacks. The most notable drawback is that position statements will only allow us to evaluate opinions against two opposing viewpoints at a time, which may be insufficient for topics that have a wider range of viewpoints. While this is an issue with our scheme, we note that it also affects some similar labelling schemes, and solving the problem in a more complete and generalisable way may require a level of computational understanding that is beyond current approaches.

1.1 Outline

The contributions of this work are in three main areas: extracting reported speech, attributing reported speech, and a new form of opinion mining. We start by discussing our work in reported speech extraction, which is covered in Chapter 2. Our main contribution in this area is that we built a fully labelled corpus that is large enough to make machine learning approaches viable. We also present four methods for quote extraction, two of which are simple baselines, and two of which are novel supervised methods. We evaluate these methods on both our corpus and a corpus by Pareti (2012), with results on both corpora showing that our best learned method is more effective than the baselines.

In Chapter 3 we move on to discuss how we can attribute quotes to speakers. The chapter starts by discussing the related work in this field, with a particular focus on how previous researchers have defined the task. The chapter then introduces the corpus that we have constructed for this task, as well as the three publicly-available corpora that we use to evaluate our methods. Next we provide a detailed explanation of how we define the task, as well as some of the alternative task definitions that are possible. This is an important contribution of this thesis as previously there was no clearly accepted definition of how the task should be defined and evaluated.

A publication completed as part of this thesis (O'Keefe et al., 2012) was the first to evaluate quote attribution with multiple corpora. In Chapter 3 we are able to expand the coverage even further to include a fourth corpus that was unavailable at the time that work was published. The chapter then proceeds to clearly define the experimental setup and the evaluation, with some necessary caveats for some of the corpora, before moving on to discuss the approaches. Our approaches build upon the work of Elson and McKeown (2010). While their method is effective, we show that it relies on the use of gold-standard features, which is not realistic in practise. We present results using various class labelling schemes and sequence decoding methods. These allow us to replace the gold-standard features with predicted values. We also vastly expand the feature set, which results in a large gain in performance. We finalise the chapter by presenting and discussing the results. This includes several experiments that replicate the setup of previous studies, which allows the first like-for-like comparison of results between quote attribution studies.

While Chapters 2 and 3 focus on establishing the task, corpora, methods, and comparisons with previous work, Chapter 4 is focused on evaluating the full pipeline that comes before opinion mining. For this evaluation the main missing component is the use of automatic mention detection and coreference resolution, which is needed to find candidate speakers. By running these systems we can get a more realistic measure of how well the full pipeline of systems performs. This presents a challenge though, as the annotations that we have link quotes to gold-standard speakers, which do not necessarily align with speakers produced by the automatic systems. To solve this issue, we define two alignment methods that allow us to align the gold-standard annotations with the automatic coreference chains. This resolves the aforementioned challenge, allowing us to run experiments with the automatic coreference chains. These results also provide some insight into how well coreference resolution systems perform, as we are in effect using quote attribution as an extrinsic evaluation of coreference resolution.

Once we have established how coreference resolution affects the task, we present an evaluation of the full pipeline of automatic coreference resolution, automatic quote extraction, and automatic quote attribution. This evaluation shows that the results depend on the alignment method used, and thus highlights the need for an extra step that further disambiguates the textual references to entities. We thus use a Named Entity Linking (NEL) system to link candidate speakers to a Knowledge Base (KB), which gives us a cross-document representation of each speaker. With this, we perform a final evaluation that answers the question of how many errors we would expect from the full pipeline of systems that precedes the opinion mining process.

Having established the efficacy of our quote extraction and attribution methods we move on to review the previous work on opinion mining in Chapter 5. This review is focused on the various tasks that are related to opinion mining, and on how the task definitions impact the meaning of their output.

In Chapter 6 we introduce our own novel definition of opinion mining and present a corpus that has been labelled according to this definition. Our definition of the task is motivated by a desire to be able to categorise arguments in a debate. As such, we define an opinion as a tuple of an opinion holder, a polarity, and a position statement, where the polarity is relative to the position statement. We then move on to describe the 6.1 that we have built for this task, as well as some of the annotation and we conclude the chapter with preliminary results in identifying opinionated quotes.

We conclude this thesis in Chapter 7 with a discussion about both the contributions and the limitations of this work. The chapter argues that by producing new data and new methods we have made the tasks of quote extraction and quote attribution much more consistent, which provides a basis for future work. On opinion mining we have defined a new task that allows us to categorise opinions that did not fit into previous task definitions. While our work does not provide novel approaches that are specially designed for this task, we have provided a definition, a corpus, and preliminary results that will form the backbone of future endeavours.

1.2 Summary of Contributions

- We build a large corpus for quote extraction and quote attribution that covers all direct, indirect, and mixed quotes

- We develop novel methods for quote extraction that produce state of the art results on two corpora

- We develop novel approaches, features, and class labelling schemes for quote attribution

- We evaluate our quote attribution methods on four corpora, which shows our best method achieving a state of the art result across three of the four corpora

- We present a new opinion mining scheme that allows us to consistently label opinions in quotes
- We build a large corpus that is labelled according to our new opinion mining scheme
- We present preliminary results in discerning opinionated quotes from neutral or irrelevant quotes

1.3 Publications Associated with this Thesis

Tim O'Keefe, Silvia Pareti, James R. Curran, Irena Koprinska, Matthew Honnibal (2012). A Sequence Labelling Approach to Quote Attribution. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing (EMNLP).

Tim O'Keefe, James R. Curran, Peter Ashwell, Irena Koprinska (2013). An Annotated Corpus of Quoted Opinions in News Articles. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL).

Silvia Pareti, Tim O'Keefe, Ioannis Konstas, James R. Curran, Irena Koprinska (2013). Automatically Detecting and Attributing Indirect Quotations. In Proceedings of the 2013 Joint Conference on Empirical Methods in Natural Language Processing (EMNLP).

Tim O'Keefe, Kellie Webster, James R. Curran (2013). Examining the Impact of Coreference Resolution on Quote Attribution. In Australasian Language Technology Association Workshop 2013 (ALTW).

2 · Extracting Reported Speech

In this chapter, we describe methods of automatically extracting direct, indirect, and mixed quotes from text. It may appear that achieving highly accurate results for this would be straightforward. Direct quotes should be trivial to extract as they are marked orthographically with quotation marks, while, given an accurate parse, indirect speech can be extracted by finding the appropriate dependent of a speech verb. However there is little previous work in the area, and as such we lack strong empirical evidence of this intuition.

(4) a. *"For 10 million, you can move $100 million of stocks,"* a specialist on the Big Board gripes. *"That gives futures traders a lot more power."*

b. *Police would only apply for the restrictions when "we have a lot of evidence that late-night noise, anti-social behaviour and alcohol-related crime is disturbing the residents of that neighbourhood"*, Superintendent Tony Cooke said.

c. Mr Walsh said *Rio was continuing to hold discussions with its customers to arrive at a mutually agreed price.*

d. The greyhound won his Healesville heat in the second fastest time of the day and is a live chance in the "straight track" 340 metres scamper.

Given a document, the task of extracting quotes involves extracting all instances of direct (Example 4a), indirect (Example 4c), and mixed (Example 4b) quotes, while ignoring instances of scare quotes (Example 4d). Most quotation attribution studies (Pouliquen et al., 2007; Glass and Bangay, 2007; Elson and McKeown, 2010; O'Keefe et al., 2012; He et al., 2013) thus far have limited their scope to direct quotations, as they are delimited by quotation marks, and so appear easy to extract. These studies have operated with the assumption that extracting the text that appears between quotation marks will yield complete quotations. The problem with this is that scare quotes and the directly-quoted part of mixed quotations will be erroneously detected, which makes the precision of these rules lower than expected.

Some studies (Krestel et al., 2008; Sarmento and Nunes, 2009; Schneider et al., 2010; de La Clergerie et al., 2011) have gone further and looked at extracting indirect and mixed quotations. However, owing to a lack of data, these studies have approached the problem with either rule-based methods or hand-built grammars. None of these studies has included a large-scale evaluation, and as yet no two approaches have been evaluated over the same data. Fernandes et al. (2011) is the only work with sufficient data to allow supervised machine learning, though their system still partly relies on rules to extract indirect and mixed quotes.

In this chapter, we describe our work in addressing these issues, in particular the lack of data. In Section 2.2, we describe the previous work on quote extraction. In Section 2.3, we discuss our work in creating a corpus of 7,990 quotations from 965 Sydney Morning Herald[1] articles. We also discuss a second corpus that we use for our experiments, which was built by Pareti (2012) using Wall Street Journal articles. In Section 2.4 we discuss the experimental setup for this task, including the baseline methods that we use. Section 2.5 describes the two machine learning approaches that we evaluate, where the first is a token-based approach using a Conditional Random Field (CRF) to predict whether each token is part of a quote, while the second is a maximum entropy classifier that predicts whether parse nodes are quotations or not. Section 2.6 presents the results of these experiments.

Parts of this chapter are based on work that was published in the proceedings of the Conference on Empirical Methods in Natural Language Processing (2013). The paper was joint work with Silvia Pareti, who provided PARC*, the verb-cue classifier, the method for learning from incomplete data, and significantly contributed to the features.*

[1] http://www.smh.com.au

2.1 Quote Types

Before we discuss the related work in quote extraction, we will cover the types of quotes and their function in some more detail.

2.1.1 Direct Quotes

(5) a. Asked about the allegation yesterday, a spokesman for Joel Fitzgibbon said: *"The minister has completed a full audit of his statements of private interests and is satisfied that they are accurate and up to date."*

b. *"My government will stand firm on this policy.*

"We believe that this is the right policy at the right time."

Direct quotes (Examples 5a and 5b) are transcriptions of speech that indicate that the writer has reproduced the speech verbatim. The speech is marked orthographically through the use of either single (' and ') or double (" and ") quotation marks, which disambiguates the extent of the text that should be attributed to the speaker. In most cases direct quotes will start and end with quotation marks, however there is also a lesser-known convention where quotes continuing over multiple paragraphs (Example 5b) include additional quotation marks at the beginning of each new paragraph that forms part of the quote.

2.1.2 Indirect Quotes

(6) a. His spokesman said *he returned the gift because its value exceeded the threshold requiring declaration.*

b. Unclaimed ashes were not a new problem: *about 20 years ago fossickers found containers of ashes dumped at a Southern Highlands tip*, Mr Perram said.

Indirect quotes (Examples 6a and 6b) are an alternative to direct quotes, where the writer reports on a speech act, but does not report on the content of the speech act verbatim. Indirect quotes are not marked orthographically, so the reader needs to judge the extent of the speech themselves. In some cases, such as Example 6b, this can be challenging, as neither the syntax nor semantics of the sentence need to indicate precisely what was said and what wasn't.

Indirect quotes are used in three main instances. Firstly, if the content of a speech act is too long, then a journalist may reword it for brevity. Secondly, speech often contains disfluencies, which a journalist may seek to remove by completely rewording a speech act. Lastly, a direct quote may require context to understand, so a journalist may reword the quote to remove the need for that extra context. In all of these cases, the communicative intent of the speaker should be maintained.

2.1.3 Mixed Quotes

(7) a. Mr Warner said *the claims were being "thoroughly investigated" but there had been no change to the department's initial findings dismissing a report of an inquiry into the Defence Minister*

b. Yet Charlie Aitken, Southern Cross's head of institutional dealing, recently lent his support to the analyst assessment of K2, noting *that he could "easily see them with funds under management of $5 billion over the next five to 10 years".*

The last type of quotes that we seek to extract are mixed quotes (Examples 7a and 7b). Mixed quotes are part direct and part indirect, that is, part of the quote appears within quotation marks, while part of it appears outside quotation marks. The direct part, is typically understood to be verbatim, while the part outside the quotation marks is usually considered to be reworded.

Mixed quotes are used in two main cases. The first is shown in Example 7a, where the bulk of the quote is indirect. In this case the quoted part is used to distance the journalist from the particular wording or claim being made. The second case is when the bulk of the quote is direct, but the journalist needs to reword some small segment of the quote to fit in with the grammatical structure of the sentence.

Authors	Method	Language	Quotes	P (%)	R (%)
Krestel et al. (2008)	grammar	English	133	74	99
Sarmento and Nunes (2009)	patterns	Portuguese	570	88	5*
Fernandes et al. (2011)	ML and regex	Portuguese	205	$64^\diamond$	$67^\diamond$
de La Clergerie et al. (2011)	patterns	French	40	87	70
Schneider et al. (2010)	grammar	English	-	$56^\diamond$	$52^\diamond$
Weiser and Watrin (2012)	grammar	French	140	78.9	-

Table 2.1: Related work on direct, indirect and mixed quotation extraction. Note that these results are not directly comparable as they apply to different languages and differ in evaluation style and test sets. The item marked with an asterisk (*) was estimated by the authors for extracting 570 quotations from 26k articles. The items marked with a diamond ($^\diamond$) are results for quotation extraction and attribution jointly.

2.2 Related Work

The first speaker attribution systems (Zhang et al., 2003; Mamede and Chaleira, 2004; Glass and Bangay, 2007) were developed for children's literature and were concerned with associating quotes with characters, so that speech synthesis systems could read each character's part in a different voice. They were concerned only with extracting direct quotes, as indirect quotes are infrequent in literature, and in any case should be read in the narrator's voice. As such, these studies extracted direct quotes with simple rules.

In extracting direct quotes, the work of Elson and McKeown (2010) is the largest study to date. As in earlier work, they looked at the problem in the narrative genre, with a focus on using large-scale data and machine learning to attribute quotes. To extract direct quotes they returned all of the text between quotation marks. They found that in their corpus of 3,578 quotations, 112 (3.5%) were non-dialogue text, which implies a baseline precision of 96.5% in extracting direct quotes. While they aimed to automatically detect these cases as part of the attribution process, they did not report separate results for this part of the task.

Other work has looked at extracting direct quotes in news (Pouliquen et al., 2007; Liang et al., 2010), with our own work (O'Keefe et al., 2012) looking at the task on a large scale.

In that work, which was completed as part of this thesis, we used simple rules, much like Elson and McKeown (2010), and found that over 99% of direct quotes were detected. Our definition of direct quotes in that work was more relaxed, as it considered scare quotes and the directly-quoted portions of mixed quotes to be valid quotes. This is despite the fact that for news in particular, the interpretation of a mixed quote can change when its indirectly-quoted portion is included, as in Example 4b. We would also expect news text to include many more scare quotes (Example 4d) than literature, as they can be used by journalists to distance themselves from a particular claim or phrasing. It is not clear that scare quotes should be attributed to other entities. Later work in this chapter demonstrates that our early result of 99% in that study is not possible when we use a more rigorous definition of what constitutes a direct quote.

While there have been several studies looking at extracting direct quotes, extracting indirect and mixed quotes has received much less interest. In the news genre, Krestel et al. (2008) developed a quotation extraction and attribution system that uses a hand-built grammar to detect whether constructions match six general lexical patterns. These patterns rely on the use of *reported speech verbs* – such as said, claimed, wrote, and so on – to identify when a quote is introduced. To evaluate their work they annotated seven articles from the Wall Street Journal, which contain a total of 133 quotations. They achieved macro-averaged precision of 99% and recall of 74% when detecting the span of the quotation. However, they did not give a breakdown of the different types of quotations (i.e. indirect, mixed, and direct), so it is unclear how challenging these articles were.

Schneider et al. (2010) introduced a system called PICTOR, which is primarily intended to visualise quotations over time. Though they focused on visualisation, they included a method for detecting direct, indirect, and mixed quotes. Their approach used a manually-defined context-free grammar (CFG), which was built with reference to a small set of labelled articles. They did not specify the exact number of articles, but they noted that their method achieved 75% precision and 86% recall in terms of words correctly ascribed to a quote or speaker, while it achieved 56% precision and 52% recall when measured in terms of completely correct quote-speaker pairs. We note, however, that this evaluation was performed using the same data that was used to construct the CFG, so their reported results are potentially better than they would be on unseen data.

Weiser and Watrin (2012) also developed a grammar-based approach. Their method relied on a list of reported speech verbs, so they first focused on creating this list. They theorised that the speech verbs used to introduce direct quotes would be much the same as the ones used to introduce indirect quotes. Using this intuition, they built a speech verb list by detecting direct quotes using rules, and then finding the verb used to introduce each quote. They then used this list of verbs to find other sentences that contain a speech verb, but no direct quote. They argue that there are sixteen patterns that can be used to introduce indirect speech. They test the sentences they detected against two of these patterns and find that they achieve a precision of between 74.5% and 78.9%.

There has also been work on extracting direct, indirect, and mixed quotes from news in languages other than English. SAPIENS, presented by de La Clergerie et al. (2011), extracts quotes from the L'Agence France-Presse[2] newswire. It finds direct quotes using simple rules, and then finds indirect and mixed quotations by parsing the text, and then finding instances of speech verbs from a pre-built list. They then use syntactic patterns to find the complement and subject of each verb, where the complement is the content of the quote, and the subject is the speaker. They evaluated their system with 40 quotes that were randomly sampled from 40 different articles. Their results show that their system was able to correctly predict the span of 28 of the quotes, while it made 4 erroneous predictions.

VERBATIM (Sarmento and Nunes, 2009) was built to work over Portuguese news articles, and uses similar methods to SAPIENS. The authors define a set of 35 speech verbs, which they search for in a target article. Any sentence containing one of these speech verbs is then checked to see if it matches one of 19 patterns that were also manually defined. As their work is intended to show which quotes are in the news, they include a corpus-wide de-duplication step, so that they should only see each unique quote once. They evaluated their system by manually examining the system's output. The system produced 570 unique quotations from 26,266 news items, which corresponds to roughly

[2] http://www.afp.com

1 unique quotation for every 46 news items. They estimate their precision to be 88.1%, as the system was incorrect for 68 of the 570 quotes.

Also intended for Portuguese news is the system presented by Fernandes et al. (2011), which is one of the few systems to use Machine Learning (ML). They built a corpus of quotes that covers 685 news articles containing a total of 1,007 quotations. They use 802 of these quotations to train an Entropy-guided Transformation Learning (ETL) classifier with Named Entity (NE) and Part-of-Speech (POS) features. Their system is a hybrid, in that it used the ETL classifier to detect the beginning of quotes, while a regular expression is used to detect the end. They evaluate their system on the remaining 205 quotations and find that they achieve 64% precision and 67% recall when jointly predicting the quotation and speaker. They do not report results on quote extraction alone.

As part of building the MPQA, Wilson (2008) define two frames that contain elements of reported speech. The first, direct subjective expressions, mark the speech verb, but do not mark the content of what was said in any way. The second, objective speech events, have a target field that represents the content, however this was not annotated. These limitations mean that the MPQA corpus is not currently appropriate for the full task of quote extraction that we discuss here.

These approaches to detecting indirect and mixed quotes are summarised in Table 2.1. The table shows that before our work (Pareti et al., 2013), the largest evaluation covered only 570 quotations, while the only publication that uses ML did not report separate results for quote extraction. In addition, there has been no study thus far that includes a comparison with previous work, which limits the conclusions that we can draw about the effectiveness of the proposed methods. In the next section we will introduce our corpus, the SMHC, which contains annotations of all direct, indirect, and mixed quotes from 965 news articles.

2.3 Corpora

The previous section demonstrates that there is a clear lack of large-scale data for this task, and as a result the relative performance of the various methods is poorly understood. In this section, we describe our work in producing the Sydney Morning Herald Corpus (SMHC), which is the first fully-annotated large-scale corpus of direct, indirect, and mixed quotations. We also discuss Penn Discourse Treebank Attribution Relations Corpus (PARC) (Pareti, 2012), which is an even larger corpus that contains many more types of annotations, though the corpus is not yet fully annotated. In later sections we present results on both of these corpora.

2.3.1 Sydney Morning Herald Corpus (SMHC)

To build the SMHC we collected 965 articles from the 2009 Sydney Morning Herald (SMH).[3] These articles were initially collected and preprocessed as part of a NEL project (Hachey et al., 2013), that also included gold-standard NE annotations. This allowed us to annotate not only the content span of each quotation, but also its speaker, which we discuss in more detail in Chapter 3. The annotation scheme was produced with reference to the scheme from Pareti (2012), and is included in this thesis as Appendix A.

For each document, annotators were asked to read through and identify all of the direct, indirect, and mixed quotations that were present. For this work, we considered quotations to include any acts of communication that include some or all of the content of what was communicated, as Pareti (2012) does. More specifically, this means that examples such as Example 8a should be annotated, as they include content, while Example 8b should not, as it only notes that a person was speaking, not what they said. This also means that scare quotes (Ex. 4d), book titles, and similar text that appears between quotation marks, but is not an act of communication, should not be annotated.

(8) a. He said "the policy would cost thousands of jobs."

b. He talked about the policy's impact on jobs.

Our initial work on this corpus was intended for quote attribution, where we annotated the speakers of direct quotes that had been found automatically. We used this set of documents as our starting point, as it meant the direct quotes were already marked. A single annotator then manually marked instances of indirect quotations, and expanded the

[3] http://www.smh.com.au

extent of mixed quotations to include the indirect parts. The annotator also removed previous annotations of quoted text that does not correspond to a direct quote in our new scheme. Lastly, the annotator marked the speaker and source of each quotation (discussed in detail in Chapter 3), although we do not mark the cue verb in this work. Since the starting point of this annotation was a set of documents with direct quotes marked, every document in this corpus contains at least one direct quote.

In total the corpus contains 7,990 quotations. We use 60% of this corpus as training data (4,872 quotations), 10% as development data (759 quotations), and 30% as test data (2,360 quotations). Early experiments were trained using the training set and were tested with the development data. The final classifiers were trained on both the training and development sets and the reported results in this chapter are on the unseen test data.

2.3.2 Penn Discourse Tree Bank Attribution Relations Corpus (PARC)

PARC (Pareti, 2012) is a semi-automatically built extension to the Penn Discource TreeBank (PDTB) (Prasad et al., 2008) which covers 2,280 articles from the Wall Street Journal (WSJ). This corpus contains annotations of Attribution Relations (AR), which are more general than just acts of communication. Pareti (2011) defines ARs as having three main elements:

Content: the span of text that is being attributed.

Source: the span of text that indicates who the content is attributed to, e.g. president Obama, analysts, China, she.

Cue: the lexical anchor of the attribution relation, usually a verb, e.g. say, add, quip.

In addition each AR has a type, which captures how the source is related to the content. The four types are:

Assertion: Mr Abbott said *that he will win the election.*

Belief: Mr Abbott thinks *he will win the election.*

Fact: Mr Abbott knew *that Gillard was in Sydney.*

Eventuality: Mr Abbott agreed *to the public sector cuts.*

While each of these types clearly attributes some act, belief, or knowledge to its source, note that only assertions imply a communicative act has taken place. Due to this we use only the assertions from this corpus, while ignoring the other types. The *content* of an assertion can be used as the quote itself, while the *source* is a textually-grounded reference to the speaker. PARC also contains nested quotes (i.e. quotes within quotes), which we do not consider in this work.

The major drawback of PARC is that it is not yet fully-annotated. The version that we use contains articles with both labelled and unlabelled quotes, which presents a challenge for both training and evaluation. The corpus does include a test set of 14 articles that are fully annotated, which we use as test data for our experiments in this chapter. The test set was manually constructed by two annotators, with agreement of 87% using the *agr* metric (Wiebe and Riloff, 2005). Using the test set, Pareti estimates that 30-50% of the ARs in the wider corpus are not annotated, which presents a major challenge for learning. In joint work with Pareti (Pareti et al., 2013), which was completed as part of this thesis, we describe methods of ameliorating this problem, which will be further described in later sections.

2.3.3 Comparison

Though both SMHC and PARC are in the news genre, it is difficult to make comparisons between them. The major issue is that although the SMHC is fully annotated, PARC is not, which makes per-document averages difficult to compare. Additionally, the articles that comprise the SMHC were chosen to include at least one direct quote, and so have an inherent bias towards articles with more quotes. These facts are made clear in Table 2.2, which shows that the SMHC has a higher density of quotes than PARC at 8.3 to 4.6 quotes per article. Even once the articles in PARC that contain no quotes have been removed (786 articles), we find that the SMHC still has more quotes (8.3 vs. 7.1), despite being shorter on average (623.3 tokens versus 678.7 tokens).

Despite the difficulties in comparison, it is apparent that the SMHC has fewer indirect quotes. This could be due to either stylistic differences between the journalistic guidelines at the SMH and the WSJ, or it could be due to a move away from indirect quotes over time,

	SMHC		PARC		PARC (no null)	
	Corpus	Per Article	Corpus	Per Article	Corpus	Per Article
Docs	965	-	2,280	-	1,494	-
Tokens	601k	623.3	1,139k	499.9	1,014k	678.7
Quotations	7,990	8.3	10,526	4.6	10,526	7.1
Direct	4,204	4.4	3,262	1.4	3,262	2.2
Indirect	2,928	3.0	5,715	2.5	5,715	3.8
Mixed	857	0.9	1,549	0.6	1,549	1.0

Table 2.2: Comparison of the SMHC and PARC, reporting their document and token counts and per-type occurrence of quotes overall and per article. The "PARC (no null)" column refers to PARC where documents containing no quotes have been removed.

as the PARC contains documents from 1989, whereas the SMHC contains documents from 2009. This result also extends to mixed quotes, with the SMHC containing 0.9 mixed quotes per document, while PARC contains 1 per document.

These results indicate that for the experiments we describe in the next section, we should expect PARC to be a more challenging corpus than the SMHC. The primary reason is that since direct quotes are marked orthographically, they should be easier to extract. In the SMHC they make up 53% of all quotes, while in PARC they make up only 30% of the quotes.

2.4 Experiments

In this section, we describe the setup of our quote extraction experiments. More specifically, we are interested in extracting the *content* of all direct, indirect, and mixed quotes within a given document. As noted in Section 2.3, we are only interested in the quotes that actually express the content of what the speaker said (Ex.8a). Text that only describes that communication took place (Example 8b) is not considered a quote in this work. We also note that for some quotes, the content being communicated is split into multiple, separate spans of text (Example 4a). To simplify our experiments, we consider each of these separate spans to be an individual quote, even though they are part of the same speech act.

2.4.1 Preprocessing

Prior to running the experiments, we performed several preprocessing steps. First we apply standard linguistic pre-processing using C&C tools (Curran and Clark, 2003). This includes sentence boundary detection, tokenisation, and POS tagging. Next we find the lemma of each token using `nltk.wordnet.WordNetLemmatizer` (Bird et al., 2009). We use the Stanford factored parser (Klein and Manning, 2002) to find both the phrase structure tree and the Stanford dependency tree. We also aim to avoid overfitting by anonymising textual references to named entities, which involves replacing them with a special string. We also normalise quotation marks to a single character, as we found that directional quotation marks were often incorrect for multi-paragraph quotes.

The output of all of the aforementioned steps, along with the document itself, is then loaded into DOCREP, which is a document representation format (Dawborn and Curran, 2014). DOCREP enables us to efficiently represent both the document and any metadata we need in a single file. It also makes development of features and error-checking much simpler than using a custom-developed data model. In the experiments in this chapter, we used gold-standard named entities in both corpora, so an additional Named Entity Recognition (NER) step would be required for documents that are not part of our corpora. In Chapter 4, we evaluate quote extraction with several automatic coreference systems.

2.4.2 Evaluation

In evaluating our system we use two metrics. The first metric, called *strict*, will only count predicted quotes as correct if they exactly match a quote from the gold standard. While this metric has the advantage of being straightforward, it tells us nothing about the predictions that were incorrect. In fact, if we took the gold standard quotes and removed a single token from each quote, every quote would be counted as incorrect, despite being very close to the gold standard. In order to avoid this issue we adopt a second evaluation metric, called *partial*, that counts partial matches as partially correct.

Strict

For the strict metric we define precision and recall as follows:

$$P = \frac{tp}{tp + fp}$$

$$R = \frac{tp}{tp + fn}$$

where tp is the number of predicted spans that exactly match a quote span from the gold standard, fp is the number of predicted spans that do not exactly match a span from the gold standard, and fn is the number of gold standard quotes that were not exactly matched by a predicted quote span. Note that this is a span-based definition of correctness, so two non-overlapping predicted spans that exactly cover a gold span will not be counted as correct, and will in fact count as two incorrect predictions. We define F-score in the usual way:

$$F = \frac{2PR}{(P + R)}$$

Partial

For the *partial* score, P and R are not defined in the usual way, and are instead defined as follows:

$$P = \frac{\sum_{g \in gold} \sum_{p \in pred} overlap(g, p)}{|pred|}$$

$$R = \frac{\sum_{g \in gold} \sum_{p \in pred} overlap(p, g)}{|gold|}$$

where $gold$ is the set of gold quotes, $pred$ is the set of predicted quotes, and $overlap(x, y)$ is the proportion of tokens in y that are overlapped by x, or more precisely:

$$overlap(x, y) = \frac{|x \cap y|}{|y|}$$

Consider the following simple example of one prediction and one gold quote, where *italics* indicates gold standard and underline indicates a prediction:

> The premier said *that his government would stand firm* and then ended the press conference.

In this example the prediction starts too late at government and erroneously includes the tokens from and forward. In this case P, R, and F for the *strict* metric would be zero, as there is no exact match. However, for the *partial* metric, the scores will be non-zero. Notice that as we are only looking at one prediction and one gold quote, we only need to work out the two overlap scores, which would be $\frac{4}{10}$ for P as there are 4 overlapping tokens and 10 predicted tokens, while for R it would be $\frac{4}{6}$ as there are 4 overlapping tokens and 6 gold tokens. This idea works similarly when we scale up the number of gold quotes and predictions. As with the strict score, we define F-score in the normal way:

$$F = \frac{2PR}{(P + R)}$$

Further considerations

Each of these metrics gives us a score for an individual article, so to find a corpus-wide result we micro-average each of these scores. Micro-averaging works by summing up the tp, fp, and fn individually from each article, before calculating the P, R, and F-scores. By contrast, macro-averaging averages the P, R, and F from each article, and so puts each article on an equal footing, rather than each quote. As we are interested in knowing how many quotes we got correct, we report micro-averaged results.

In some sections, we also report results for the different types of quotes, i.e. direct, indirect, and mixed. This raises a further issue with evaluation. If the system makes an incorrect prediction that still partially overlaps a gold quote, then the predicted quote might have a different type to the gold quote. For example, suppose that the system missed the indirect portion of a mixed quote, and only found the direct part. Should this count as a false positive mixed quote, or a false positive direct quote?

We resolve this problem differently (though analogously) for P and R. For P we restrict the set of predicted quotes to only those predictions with the requisite type, while still considering gold quotes of all types. This means that for P we will count any predicted quote of the right type as incorrect if it does not match a quote in the gold standard, which is the usual definition. So for the given example, the predicted direct quote overlapping the gold mixed quote would count as a fp for our direct quote P score. When calculating R, we restrict the set of gold quotes to only those with the requisite type, which produces an analogous argument for correctness. This applies to both the strict and partial scores.

2.4.3 Baselines

For this task we propose three baseline methods, which allows us to compare our approaches and to assess the difficulty of the task. By doing this we can assess whether our methods really do boost performance, whilst also showing what can be expected from rule-based approaches. The three baselines are: B_D, which extracts direct quotes by returning content between quotation marks; B_R, which extracts indirect and mixed quotes with a simple lexical rule; and B_S, which uses the dependency parse to find the object of each speech verb. B_R and B_S do not perform well on direct quotes, so we augment their predictions with the predictions from B_D. In cases where their predictions overlap, we discard the shortest overlapping quotes until there are no more overlaps.

Direct (B_D)

Our first baseline is to simply extract all of the direct quotes. For the vast majority of cases this involves finding the spans of text that appear between quotation marks, as in the following example:

> *"My government will stand firm"* said the premier.

While the above example accounts for most cases, there is a lesser-known convention used by some journalists when a quote continues over multiple paragraphs. It involves adding an additional open quote mark at the beginning of each paragraph after the first, with a final closing quote mark at the end of the quote. For example:

> *"My government will stand firm on this policy.*
>
> *"We believe that this is the right policy at the right time."*

These cases are also fairly simple to extract, however failing to account for them can break rule-based direct quote extraction systems.

The other issue with this baseline is that it can produce false positives by returning scare quotes, titles, and other items between quotation marks that are not speech. We address this issue by removing any prediction that has three or less tokens, and by removing any prediction that has only title-cased words and stopwords. While this approach does remove some correct predictions, the drop in recall is small compared to the gain in precision.

Lexical (B_R)

Our next baseline is a simple rule-based approach to extracting direct, indirect, and mixed quotes that uses no linguistic knowledge. It finds any tokens whose lemma matches one of the verbs from the list of speech verbs built by Krestel et al. (2008). Once it has found such a verb it returns the longer part of the sentence on either side of the verb as a quote. Consider the example from the previous section:

> The premier said that his government would stand firm and then ended the press conference.

In this case the verb said is the third token, with two tokens to its left and twelve tokens to its right. As the right hand side is longer, B_R would predict that the following is a quote: that his government would stand firm and then ended the press conference.

Syntax (B_S)

The final baseline is inspired by various methods in the literature that use the syntactic structure of sentences to find quotes. Similarly to B_R, this method first finds any token whose lemma matches a verb in the list from Krestel et al. (2008). The difference is that rather than simply finding the longest span, this method instead looks for the object of the verb, found using the Stanford dependencies. While the Stanford dependencies have a hierarchy of types with several types that potentially correspond to the object, we simply take the *ccomp*, the clausal complement, as we found that it produced the best results on our development data. This baseline is close to the method in de La Clergerie et al. (2011).

Figure 2.1 shows an example, where B_S would return: that his government would stand firm. It does this by finding the *ccomp* of the speech verb said, and then returning the *ccomp* and all of its dependants. The remainder of the sentence, from and forward, would not be returned as part of the quote.

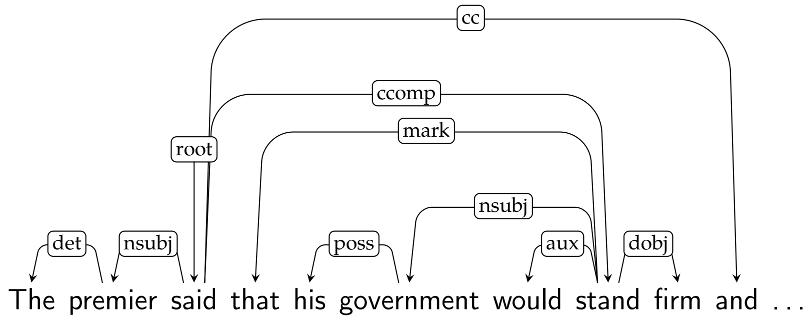

cc
ccomp
mark
root
nsubj
det
nsubj
poss
aux
dobj
The premier said that his government would stand firm and ...

Figure 2.1: B_S would extract stand and all of its dependants. The dependency tree was taken directly from the output of the Stanford parser.

2.5 Supervised Approaches

In this section, we present two supervised approaches to quote extraction that treat the task quite differently. The first is a sequence labelling approach that is inspired by the methods commonly used in NER, POS tagging, and similar tasks. The basic idea is to treat the tokens as a sequence, where each token has some observed features and a hidden label that must be predicted. The second approach is to label each node in the syntactic parse tree as being a $quote$ or $\neg quote$, using a binary classifier. Similarly to the baseline methods, both the token-based approach and the constituent-based approach use the predictions made by B_D. Overlapping predictions are resolved by greedily selecting the predictions with the longest span. We found that this improved the performance of both approaches, although the token-based approach only showed a minor improvement. Although the item being classifier is fairly different, i.e. a token vs. a parse node, we have kept the features fairly consistent between the two models, so that the relative strength of the two approaches can be assessed.

Before we move on to discuss the features, it is worth discussing why we cannot use a parser more directly for this task. By adding an extra label to parse nodes that represent quotes in the training data for a parser, we could train one of a number of parsers to detect quotes, similarly to how Favre and Hakkani-Tür (2009) use a parser to detect apposition. While this method can be applied equally as well in theory, in practise we do not yet have the data, as we need both gold standard parses and gold standard quotes. The PARC has gold standard parses available, but is not yet fully annotated, while the SMHC is fully annotated but has no gold standard parses. In our view, the method that we describe to avoid the unlabelled data in PARC would not work nearly as well with a parser, as it would miss out on too much data. Once PARC is fully annotated it would be well worth investigating using a parser for quote extraction.

2.5.1 Common Features

The common features can be separated into several distinct feature sets. All of the features are calculated with respect to a target, t, which is either a parse node or a token, depending on whether the features are used for the constituent- or token-based approach.

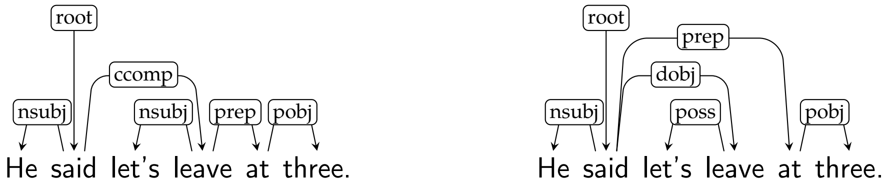
Lexical

The lexical features are intended to encode t itself, as well as the lexical context that t is in. For the constituent-based approach there will not be one specific token that we can use as the target, so we treat the first and last tokens in the constituent as two separate targets, and generate features relative to each of those tokens. The lexical features are unigram and bigram versions of the token, its lemma, and its POS tag. These are calculated for each of the tokens (or pairs of adjacent tokens) in a window from five tokens preceding t to five tokens following t.

Sentence

We include sentence features so that the learners have some evidence with which to determine if t's sentence contains any attributable text at all. These features look for clues that are usually associated with quotes, such as quotation marks, named entities, speech verbs, and pronouns. We generate indicator features for each of these clues independently and for every combination of these clues. The speech verbs come from a small set that we manually created.

In addition to these features, we have two indicators, one for whether t is within a direct quote as detected by B_D, and another for whether there is a speech verb in the last three tokens of the sentence. The first of these prevents the systems from erroneously



(a) Example with at attaching to leave.

(b) Example with at attaching to said.

Figure 2.2: Example of the two interpretations of a difficult attachment decision.

ending a quote when there is clear evidence from the quotation marks that it should continue. The second helps to start either of the approaches labelling quotes when there is a speech verb towards the end of the sentence.

Dependency Parse

The dependency features make use of Stanford dependency parses (Klein and Manning, 2002), which can provide evidence for where some difficult quotes end. In many cases the lexical cues will provide little evidence for the end of a quote, particularly when the token following the quote is a preposition with a difficult attachment decision.

We demonstrate this with an example, shown in Figure 2.2. In the figure there are two possible interpretations of the same sentence. In Figure 2.2a at and its children are attached as descendants of the object of the speech verb (said), which indicates that they should be interpreted as being said by the speaker (He), and thus should be included as part of the quote. By contrast, in Figure 2.2b, at is attached higher up in the tree, which indicates that the speaking was done at three. While this example constitutes a difficult case for a dependency parser, it is worth noting that the lexical features provide no evidence either way. However, if we assume that we have an accurate dependency parse then we can use the information it provides to correctly classify this example.

In order to capture this information we generate several features. For the token-based approach these features can be generated relative to the token, whereas for the constituent-based approach they are generated for the token in the constituent's span that is highest in the dependency tree. The features are the relation tag between t and its head, the lemma of t's head, the concatenation of those two features, and versions of the preceding three features lexicalised with t's lemma.

External Knowledge

In some cases there will be named entities with an apposition that mentions their role, as in the following example:

> Clearly, said John Howard, former Prime Minister, we should be lowering taxes...

In these cases it can be helpful to provide extra evidence that the entity or role should not be included as part of a quote. To do this we make use of three lexicons: one of roles, one of organisations, and one of titles. The role and organisation lists (Radford and Curran, 2013) were built by recursively following the WordNet (Fellbaum, 1998) hyponyms of person and organization respectively, while the title lexicon was built by hand.

This external knowledge is encoded into features by searching for any tokens that match an entry in one of the lexicons and then generating an indicator feature that is position-indexed relative to t. In the case of the constituent-based approach the position of t is taken to be the first token within the span of the constituent. The indicator features are generated as separate features for the different lexicons.

2.5.2 Token-based Approach

In the token-based approach, we treat the task as a sequence labelling problem, where we have a sequence of tokens, with some features, and we want to predict a label for each token. This task can be accomplished by using a binary classifier on each token independently, however this will not necessarily result in a good sequence of decisions. As such, it is much more common to use a sequence labelling approach such as a Hidden Markov Model (HMM), a Maximum Entropy Markov Model (MEMM), or a Conditional Random Field (CRF). For our experiments we use a first-order linear chain CRF[4] (Okazaki, 2007), as it does not suffer from the label bias problem (Lafferty et al., 2001) and tends to outperform the other two methods for tasks with a moderate number of classes.

[4] http://www.chokkan.org/software/crfsuite/

The	premier	said	that	he	would	stand	firm	and	then	ended	the	press	conference
O	O	O	B	I	I	I	I	O	O	O	O	O	O

Figure 2.3: Example of IOB notation. B indicates the beginning of a quote, I indicates the token is within a quote, and O indicates the token is outside a quote.

For the labels that we are trying to predict we use the popular Inside-Outside-Begin (IOB) labelling scheme. In IOB notation, each token is labelled with B if it is the beginning of a quote span, I if it is within a quote span, but not the first token in the span, and O if it is not part of any quote span. An example of IOB labelling for this task can be seen in Figure 2.3.

For most sequence labelling tasks, such as NER and POS tagging, the sequence that we are trying to predict will be a whole sentence. The reason for this is that we expect any cross-sentence dependencies to be relatively minor, when compared to the in-sentence dependencies. In our task, however, it is valid to have a quote that covers multiple sentences, so we need to take into account a wider sequence. As such we use whole paragraphs as the sequence, as the vast majority of quotes are less than a paragraph long, and the cases that are longer are direct quotes, which can be detected with rules.

For features we use both the common features, where the target of each feature is a token, as well as some features that are specific to the token-based approach. These specific features are intended to be counterparts to the additional features in the constituent-based approach. All of these features are calculated relative to a target token, t.

Verb

The verb features capture the relationship between t and any speech verb that may be its syntactic ancestor. There is an indicator feature for whether t is a descendant of a speech verb, and another for whether t is the first token in a constituent that is a descendant of a Verb Phrase (VP), where the main verb is a speech verb.

Ancestor

The parse nodes that have t in their span may provide some evidence about whether t should be included in a quote or not. As such, we generate features with the node label of every parse node that contains t in its span, where each of the features is indexed by their depth in the parse tree.

Syntactic

We calculate the final set of features relative to the highest constituent that has t as the first token in its span. If t is at the beginning of a quote, then in the constituent-based approach we would label this node as a positive instance. As such, if we generate features relative to this node then we can directly capture some of the information that is available to the constituent-based approach. In Figure 2.4, if we consider t to be exports, then the node would be the $SBAR - Q$ node.

We generate features for both the node mentioned above, as well as its parent, if it has one. The features are the label, depth, and token span size of both of these nodes, and an indicator feature for whether either of the nodes contains a speech verb. These features should help to distinguish the beginning and end of quotes.

2.5.3 Constituent-based Approach

The constituent-based approach is based on the idea that the span of each quote should be represented by a constituent in the parse tree. In most cases, this constituent should be the child of a verb phrase, where the main verb of the verb phrase is a speech verb. Using this idea we should be able to use a classifier to decide whether each parse node represents a quote or not. This idea is set up in our constituent-based approach as a binary classification task, where parse nodes are labelled as either $quote$ or $\neg quote$. This is shown in Figure 2.4.

Ideally each quote should be contained in exactly one constituent, with no tokens of the quote being part of other constituents, and no tokens within the parse node that are not part of the quote. Unfortunately this is often not true for direct or mixed quotes de-

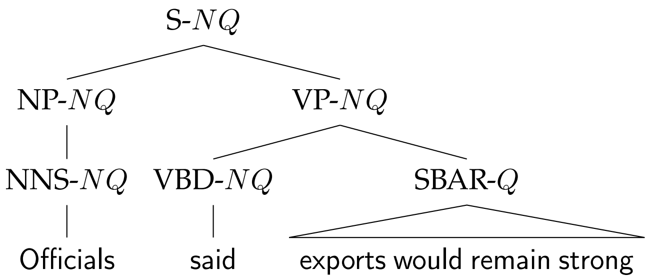


Figure 2.4: The simple case for the constituent-based approach where exactly one constituent matches a quote. Q represents a $quote$ while NQ represents $\neg quote$.

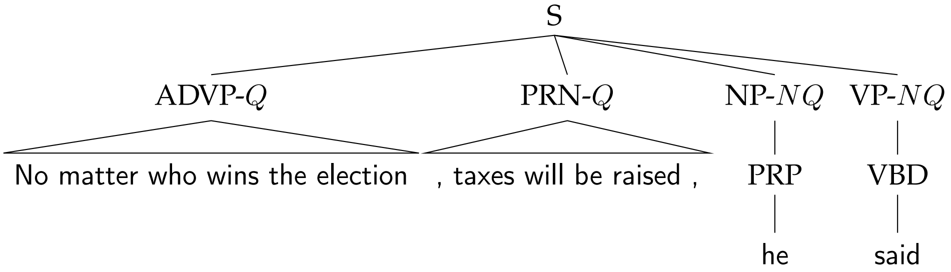


Figure 2.5: Two constituents forming a single quote, so both are labelled with Q, as there is no higher-level node that exactly matches the whole quote. Q represents a $quote$ while NQ represents $\neg quote$.

pending on where they start. As such, we represent these cases by labelling as a $quote$ every node whose projection is a subspan of the quote, but whose parent's projection is not a subspan of the quote. For these cases there will be multiple positively-labelled nodes that correspond to a single quote, so we need to introduce a post-processing step that recombines the predictions into a single quote. We do this by joining any adjacent or overlapping $quote$ predictions that are within the same sentence. By restricting the merging process to be within a single sentence, we remove the potential for the constituent-based approach to make predictions that cover multiple sentences. However in experiments on the SMHC development set we found that allowing quotes to merge across sentences reduced performance.

The classifier that we use for this method is the Scikit Learn[5] (Pedregosa et al., 2011) implementation of a maximum entropy classifier using $L1$ regularisation. We only train

[5] http://scikit-learn.org/

this method using indirect and mixed quotes, as we found that it performed poorly on the development set when trained with direct quotes. In the common features listed previously, each parse node plays the role of the target for the features, except for the lexical features where we use both the start and end tokens of the node's projection. In addition to those features, the classifier uses several features that are described below.

Node

The node features are intended to provide information about the constituent itself, which classifiers can use to decide whether the constituent should be classified as a quote or not. These features are the label of the constituent, the number of descendant nodes it has, the number of ancestor nodes it has, and the number of direct children it has.

Span

The span of a constituent, i.e. the tokens that are its descendants, can be helpful in determining whether it should be classified as a quote or not. In particular, it can be useful to know whether the span contains a speech verb or any named entities. As such, we generate features for the length of the span, and indicator features for whether there is a speech verb or a named entity within the span.

Context

In order to mimic the kind of context information that is more directly available to the token-based approach, we generate features for the parent and any siblings of the target constituent. These features include all of the features from the dependency, node, and span feature sets, which are generated with respect to the parent or sibling in question, rather than the target constituent.

2.6 Results

The results of our experiments can be divided into two main areas. Firstly, we report results on direct quotes only, and demonstrate why the task of extracting direct quotes

Corpus	Method	Strict			Partial		
		P	R	F	P	R	F
PARC	B_D	75	94	83	96	94	95
	Token	97	91	94	98	97	97
SMHC	B_D	87	93	90	98	94	96
	Token	94	90	92	99	97	98

Table 2.3: PARC and SMHC results on direct quotes. The token-based approach is trained and tested on all quote types, with only the results on direct quotes shown.

is harder than it may appear. Secondly, we discuss the performance of our methods in extracting indirect and mixed quotes.

2.6.1 Direct Quotations

In a publication completed as part of this thesis (O'Keefe et al., 2012), we claimed that extracting direct quotes was trivial. While this claim was valid with the definitions in that work, it is not as clear cut when we use more rigorous definitions. Table 2.3 shows that when using a rule-based system (B_D) that can handle multi-paragraph quotes, titles, and scare quotes, we only achieve a strict F-score of 83% on PARC and 90% on SMHC. The primary reason for this is that B_D predicts that the directly-quoted portion of mixed quotes are direct quotes. For some applications these predictions may not be considered incorrect. However, it is clear that these predicted direct quotes do not exactly match the span of a quote in the gold standard, so B_D receives a low precision score for making these predictions.

By comparison, we train the token-based approach on all direct, indirect, and mixed quotes. We then automatically filter its predictions so that only the direct predictions remain. This means that it is able to avoid predicting mixed quotes, which means that it does not do nearly as badly in terms of precision. In terms of strict recall we would expect B_D to get very close to 100%. The reason it is lower is that we added restrictions to avoid titles and scare quotes, which sometimes prevents the method from finding legitimate quotes. These trade-offs were found to improve the overall performance on our development data.

With the partial score the difference between P, R, and F is smaller, as B_D does not get penalised as badly for only predicting part of each mixed quote. However, even with the partial score, the token-based approach achieves better results. Note that the token-based approach actually has a worse strict recall, which indicates that it is most likely identifying most of the direct quotes, but is occasionally getting the spans wrong, as it may predict that some direct quotes are mixed quotes.

Overall there are two interesting outcomes from these results. The first is that identifying direct quotes is not trivial, as there are issues with getting very high scores for either precision or recall. Notice however, that if you wanted 100% recall and good precision, you can turn off the rules that avoid titles and scare quotes. In terms of precision it is more difficult to achieve perfect results, however if you consider part of a mixed quote to be a valid direct quote, then you can get close to 100% precision by avoiding titles and scare quotes aggressively.

The second main point here is that B_D is not in any way incompatible with the other methods that we propose. Most of its poor precision with the strict score is due to it extracting the directly-quoted portions of mixed quotes, which get counted as incorrect. If it is used in conjunction with a method that can extract mixed quotes, then this problem no longer applies, as the system will either predict the mixed quote, or B_D will predict part of it. In terms of direct quotes, the token-based approach clearly performs well, however the constituent-based approach cannot identify direct quotes that are not introduced with a speech verb, and so it will benefit from the addition of B_D. These arguments, in addition to empirical results on the SMHC development set, justify our decision to apply B_D along with each or our proposed methods.

2.6.2 Indirect and Mixed Quotations

Tables 2.4 and 2.5 show the results of our methods on SMHC and PARC respectively. The first interesting result is that even our worst baseline, B_R, is still able to get strict F-scores of 50% and 45% with fairly balanced P and R on SMHC and PARC respectively. This

		Indirect			Mixed			All		
		P	R	F	P	R	F	P	R	F
Strict	B_R	37	42	40	15	36	21	50	50	50
	B_S	63	49	55	67	36	47	82	72	76
	Token	69	53	60	80	91	85	82	75	78
	Constituent	54	49	51	64	42	51	77	72	75
Partial	B_R	52	68	59	87	77	82	77	84	81
	B_S	75	59	66	89	66	76	91	80	85
	Token	82	67	74	88	84	86	92	86	89
	Constituent	77	63	69	91	75	82	91	82	86

Table 2.4: Quote extraction results on SMHC. *All* reports the results over all quotations (direct, indirect and mixed).

		Indirect			Mixed			All		
		P	R	F	P	R	F	P	R	F
Strict	B_R	34	32	33	17	26	20	46	44	45
	B_S	78	46	58	61	40	49	80	63	70
	Token	66	54	59	55	58	56	76	70	73
	Constituent	61	50	55	50	38	43	70	64	67
	Constituent$_G$	66	42	51	68	49	57	76	62	68
Partial	B_R	56	66	61	78	79	78	73	79	76
	B_S	89	58	70	88	75	81	92	74	82
	Token	79	74	76	85	90	87	87	86	87
	Constituent	78	67	72	84	82	83	86	80	83
	Constituent$_G$	80	54	65	90	80	85	90	74	81

Table 2.5: Quote extraction results on PARC. *All* reports the results over all quotations (direct, indirect and mixed). Constituent$_G$ shows the results for the constituent model using a gold-standard parse.

indicates that extracting direct quotes and the larger of the spans on either side of a speech verb will get about half of all quotes *exactly* right. Furthermore the partial F-scores of 85% and 82% show that B_R is getting most of each span most of the time.

B_S operates in a similar way, however it extracts the *ccomp* of the speech verb, rather than simply taking all of the text to either side. This change results in a large boost in performance. In fact, while B_R achieves strict F-scores of 50% and 45%, B_S achieves 76% and 70% on SMHC and PARC respectively. The results from these two baselines show that a large part of the task involves accurately detecting speech verbs and their arguments.

Despite the strong baseline performance of B_S, the tables show that the token-based method achieves the best results. On PARC the token-based method achieved the highest strict F-score for indirect (59%) and all quotes (73%), but was beaten by the constituent method using gold standard parses for mixed quotes. On SMHC the token-based method achieved the highest strict F-score for all quote types, with an overall F-score of 78%.

While the results across both corpora are fairly consistent for indirect quotes, there is a large difference between the corpora in how the token-based model performs for mixed quotes. On PARC the token-based approach gets P, R, and F of 55%, 58%, and 56% respectively, while on SMHC these results rise to 80%, 91%, and 85%. This difference is not seen for other methods, so we believe that these results are due to the fact that all of the quotes containing quote marks are labelled in SMHC and not in PARC. In SMHC quote marks would become very strong indicators that a quote is present, while in PARC they may not be as strongly weighted. Note that neither of the baselines can take advantage of this fact, and though the constituent model could, it is more difficult as quotation marks often sit outside the phrase nodes that would otherwise represent quotes. In analysing this problem we noted that there were many cases where the constituent model had predicted that the quotation marks surrounding a node were quotes, but not the content of the node itself. This problem could be addressed with a more sophisticated model.

The strong performance of B_S shows us that the accuracy of the parser we use can have a large impact on the performance of our system. To investigate this effect we ran experiments on PARC using both gold standard parses from the Penn TreeBank (PTB) (Marcus et al., 1993) and the automatic parses from the Stanford parser. In terms of strict F-score, the gold standard parses produced a small improvement of 1% over the automatic parses, but surprisingly with the partial F-score the gold standard parses actually performed worse, achieving 81% to 83% for the automatic parses. This difference was driven by recall which was 74% for the gold parses and 80% for the automatic parses. In terms of precision, there was an increase when switching to the gold parses of 4%. This indicates that the constituent-based approach can learn to detect quotes in either case, but tends to make more predictions when it is given the automatic parses.

2.6.3 Tokens vs. Constituents

One of our goals when designing features for the token and constituent models was to make them largely comparable. Most of the features that we defined make sense in both models. However, for cases where features only make sense for one method, we defined matching features that would provide the same information for the other method. This allows us to compare the effect of the two different learning methods and class labelling schemes. Tables 2.4 and 2.5 show that the token-based approach achieves higher P, R, and F for both metrics across both corpora, except for a partial P that is slightly lower than the constituent model on SMHC. The token-based approach is clearly doing well quantitatively, and we found that qualitatively on the SMHC development data it was generally making sensible predictions. The one consistent exception was that it would usually fail when the speech verb and speaker were introduced in a parenthetical clause, as in the following example:

> *Finding lunar ice*, said Tidbinbilla's spokesman, Glen Nagle, *would give a major boost to NASA's hopes of returning humans to the moon by 2020.*

While the token-based model mostly produced sensible predictions, we found that the constituent-based model, despite being only slightly worse quantitatively, was qualitatively making much worse predictions. In cases where the quote aligned to a single constituent, the constituent method would generally get the correct prediction. However it would often fail when it needed to positively label two or more nodes in order to predict a quote. In many of these cases the model would predict that nodes covering a single token were quotes, or that the quotation marks surrounding a quote were quotes, but not the interior content. In other circumstances it would predict that the second half of a conjunction was a quote, without the first half, which was nearer to the speech verb. These predictions would still get the method some points using the partial metric, but they are of little practical use. We propose that in order to more fully exploit parsers for this task we would need to introduce a decoding step that forces the nodes to be labelled consistently.

The other challenge for the constituent-based method is that it has a very large class imbalance, as there are many more nodes that do not represent quotes, than nodes that do. We experimented with reducing the number of negative nodes by eliminating sentences that were unlikely to contain a quote, but we found that these experiments resulted in a drop in recall.

2.7 Summary

In this chapter, we have introduced a fully-labelled corpus for quote extraction that contains direct, indirect, and mixed quotes, and we have outlined two ML approaches to the task. We were able to run experiments on both our corpus, the SMHC, as well as a corpus from Pareti (2012) that is not yet fully labelled (PARC). By only considering sentences that contain a speech verb we were able to effectively train classifiers on PARC, even though it is not fully labelled.

Our results show that with a rigorous definition, direct quotes can be much harder to extract than what might be expected. This is due to the presence of titles, scare quotes, and mixed quotes, each of which amounts to a significant number of text spans that appear between quote marks. This difficulty can be largely resolved through the use of extra rules that account for title casing and very small quotes, and crucially, by using methods that are able to extract the full span of mixed quotes.

On the full task of extracting all direct, indirect, and mixed quotes, our results show that a token-based approach using a CRF trained over IOB labels outperforms a binary maximum entropy classifier that predicts whether the span of a parse node represents a quote. On the SMHC, the CRF approach achieved an overall F-score of 78%, while the constituent-based method achieved 75%. We propose that this is due to three main reasons: the first is a large label imbalance in the constituent model; the second is the lack of a decoding step that forces labels to be consistent; and the third and most critical reason is that parse nodes do not always align with quote spans. We expect that the constituent model would be outperformed by using an automatic parser with modified training data for this task. Unfortunately this approach will not be possible until there is a complete corpus with both gold-standard parses and gold-standard quotes available.

There are many applications where knowing the extent of quotes would be useful, particularly for tasks like fact extraction, where we may want to avoid quoted content as it is often more subjective. For these applications, the methods and corpus that were introduced in this chapter would be of great use. For our goal of opinion mining we need to know both the extent of each quote, and who said it. As such, we need methods that can link quotes to speakers. We will introduce methods capable of performing this task in the next chapter.

3 · Attributing Reported Speech

In the previous chapter we discussed methods of extracting reported speech. However, for this extracted speech to be useful for many applications of opinion mining, we need to attribute each instance of speech to a speaker. In many cases, such as in Example 9a, there will be a simple relationship between the speaker and the speech, where the speaker is the subject of a speech verb and the speech is the object. While these cases can be attributed fairly simply by a rule-based system, there are many cases that are not as clear cut, such as the quote in Example 9b, which is only attributable via an implicit link between the speech ("recess until...") and the speaker (the broadcaster).

(9) a. ["It doesn't seem the numbers are there yet, but I will continue to build my case,"]$_{quo}$ [Senator Xenophon]$_{spk}$ [said]$_{cue}$.

b. [Sandilands]$_{spk}$ [told]$_{cue}$ Austereo on Sunday [that he was unable to perform his duties on-air]$_{quo}$, prompting [the broadcaster]$_{spk}$ to put his show into ["recess until we have completed an across-the-networks review of the principles and protocols of our interaction with our audience"]$_{quo}$.

A further challenge for quote attribution systems is in deciding what exactly should be returned by a system. The simplest answer is to return the span of text that references the speaker, which would be Senator Xenophon in Example 9a. However this is uninformative in cases such as Example 10a, where the quote is attributed to a pronoun, and potentially unclear for cases where the reference is a common noun (Example 10b). While simply attributing a quote to a textually grounded mention may be helpful for some tasks, it is not enough for the opinion mining task that we describe in this thesis.

(10) a. [“The Australian Government makes no apology whatsoever for deploying the most hardline measures necessary to deal with the problems of illegal immigration into Australia,”]$_{quo}$ [he]$_{spk}$ [thundered]$_{cue}$ this week.

b. [“It doesn't matter,”]$_{quo}$ [the Premier]$_{spk}$ [said]$_{cue}$.

Given these issues, we define quote attribution as the task of finding the coreference chain that represents the speaker of a given quote. One drawback of this definition is that the quote attribution system can correctly link a quote to a textual mention, but be considered incorrect because of incorrect coreference. Similarly, if the coreference is incorrect, an attribution can incorrectly link a quote and mention, but be considered correct, as the errors could cancel out. Despite this, the definition allows us to more accurately examine how well a given quote attribution system will work under realistic conditions.

In this chapter, we will describe and evaluate our approaches to quote attribution. In Section 3.1, we will discuss the previous work on quote attribution, and provide a comparison of results. Section 3.2 describes the four corpora that are available for this task, including the SMHC, which was created as part of this work. Section 3.3 describes the experimental setup, while Sections 3.4 to 3.6 describe our approaches to this task. In Section 3.7, we will present the results of our evaluation.

Parts of this chapter are based on work that was published in the proceedings of the Conference on Empirical Methods in Natural Language Processing (2012) and the Australian Language Technology Workshop (2013).

3.1 Related Work

In Chapter 2, we discussed the literature on extracting quotes, while noting that many studies cover both quote extraction and quote attribution. In this section, we will cover the quote attribution aspects of these studies, and review other work on attribution alone.

System	Method	Domain	Language	Quotes	Types	Results
Zhang et al. (2003)	Rules	Children's stories	English	562	D	A: 47.6% - 86.7%
Mamede and Chaleira (2004)	Decision tree	Children's stories	Portuguese	35	D	A: 65.7%
Glass and Bangay (2007)	Parsing with rules	Children's stories	English	12,221	D	A: 79.4%
Pouliquen et al. (2007)	Pattern matching	News	Multilingual	120	D	P: 99.2% R: low
Sarmento and Nunes (2009)	Pattern matching	News	Portuguese	570	D, I, M	P: 98.2% R: low
Schneider et al. (2010)	Manual grammar	News	English	Small	D, I, M	P: 56% R: 52%*
Elson and McKeown (2010)	Machine learning	Literature	English	3,126	D	A: 83%◇
de La Clergerie et al. (2011)	Dependency rules	News	French	40	D, I, M	A: 53%
O'Keefe et al. (2012)	Machine learning	News & Literature	English	12,238	D	A: 53.3% - 92.4%
He et al. (2013)	Machine learning	Literature	English	1,901	D	A: 74.8% - 82.5%
Pareti et al. (2013)	Machine learning	News & Literature	English	18,517	D, I, M	A: 69% - 87%

Table 3.1: Related work on attributing quotes. These results are not directly comparable, as only three studies have presented results on public test sets. The number of quotes includes both training and evaluation sets for machine learning approaches. The type of quotes indicates direct (D), indirect (I), or mixed (M). The results marked with an asterisk (*) are for quotation extraction and attribution jointly. The results marked with a diamond (◇) use gold standard information.

Early work in quote attribution was generally focused on children's literature, with the intention of informing speech synthesis systems, so that they could read different parts in different voices. Zhang et al. (2003) propose a rule-based approach, which returns the first NE preceding the quote if it is in the same paragraph, or failing that, the first NE after the quote. They find that this is between 47.6% and 86.7% accurate, depending on the particular document.

Mamede and Chaleira (2004) expand on the approach of Zhang et al. by assigning speakers using a hand-crafted decision tree based on five rules. They evaluate their work on Portuguese children's literature, and find that their approach achieves 65.7% accuracy on a small corpus of 35 utterances. However, this result cannot be compared to Zhang et al.'s work, as the evaluation is over different data.

Glass and Bangay (2007) propose an approach that uses naïve scoring techniques and simple rules to identify the speaker of each quote. Their system works in three main stages: first they identify speech verbs; then they identify the actor of each speech verb; and lastly they resolve the actor to a speaker. The first two stages use scoring techniques with simple features to identify speech verbs and actors, while the third stage uses rules that depend on whether the actor is a pronoun, common noun, proper noun, or not present. Their approach yields 79.4% accuracy on a corpus of manually annotated children's stories. While this corpus would be useful to compare our results on, it was unavailable at the time of writing.

The other domain that studies have focused on is news. Pouliquen et al. (2007) present a news monitoring system that works using a pattern matching approach that is multilingual and able to work over tens of thousands of news articles per day. As the quotes that their system analyses are likely to appear multiple times across news providers and languages, they design a very high precision system, with the resulting low recall largely ameliorated by the inherent redundancy of the data. Their system works by finding elements in the text, such as quotation marks, person names, and speech verbs, and then checks whether they match any of a set of patterns from a manually defined list. As they are focused on high precision, their system only attempts to identify quotes spoken by a known set of 50,000 speakers, and only when they are referenced by one of a known set of aliases. As such they achieve precision of 99.2% when assigning speakers.

Sarmento and Nunes (2009) proposed a pattern matching system that works over a large number of Portuguese news articles. Their system is conceptually similar to the work of Pouliquen et al. (2007), but is able to use more precise patterns as it is focused on only one language. As we noted in Chapter 2, their system only extracted 570 quotes from 26,266 news articles, which suggests that it has low recall (if the rates of quotation per news article are at all similar to English, see Tble 3.2). However, they achieved a precision of 98.2% in identifying the speakers of the quotes that they extract.

Schneider et al. (2010) developed PICTOR, which is intended to extract, attribute, and visualise quotes from English-language news articles. To do this they manually constructed a grammar with reference to a small development corpus. Their grammar is able to identify both direct and indirect quotes, as well as their speakers. With a permissive evaluation metric that allowed for partial matches, their approach yielded 75% precision and 86% recall, however when considering exact matches they achieved 56% precision and 52% recall. It is worth noting that they did not provide separate accuracy results for solely attributing speakers, so both of these metrics considered whether both the quote span and speaker were correct.

In similar research, de La Clergerie et al. (2011) developed SAPIENS, which identifies and attributes quotes in French news articles. They identify most quotes and speakers by finding speech verbs that head the main clause of a sentence. They can then take the object of the speech verb as the quote and the subject as the speaker. Cases where quotes are introduced without a speech verb are handled by looking for prepositional phrases that can introduce speech (e.g. "according to"), and then by returning the noun phrase following the preposition as the speaker. They perform a limited evaluation of the system on 40 quotes, and find that in 12 cases their system found no speaker, while in 7 cases the speaker was incorrect.

Elson and McKeown (2010) (hereafter E&M) were the first to use machine learning and a large-scale corpus for quote attribution. They constructed their corpus from excerpts of 11 classic 19th century works of literature by six well-known authors. The corpus contains direct quotes annotated with their speakers, as well as a set of candidate speakers that were identified using the Stanford NE tagger (Finkel et al., 2005), and common noun candidates that were identified using a method outlined in Davis et al. (2003).

As part of their preprocessing they assign each of the quotes to one of several categories, which correspond to how the quote is textually attributed to its speaker. Some of these categories, such as when there is a quote followed by a speech verb and a named entity, indicate who the speaker is so heavily that it can be returned without any further processing. For the remaining quotes their system finds up to 15 candidate speakers, who are assigned a probability of being the speaker by a binary classifier. They then propose several methods for reconciling these independent classifications. Their evaluation shows that for the category-based predictions they achieve between 93% and 99% accuracy, while the accuracy on the remaining quotes is between 63% and 64%, with an overall accuracy of 83% across all quotes. This compares favourably with a baseline that assigns quotes to the nearest candidate, which achieves an accuracy of 52%. Though these results are encouraging, they rely upon gold-standard information to generate some features, so they are not realistic.

Work by He et al. (2013) extended E&M's work in several ways. Firstly, although they note that aspects of the task make it difficult to treat as a true sequence labelling problem, they build a proxy sequence model by incorporating the full set of features from the quotes adjacent to a target quote. This allows them to partially capture the sequence information in their main learning method, *SVM-rank*. Secondly, they introduce several new features, notably including an actor-topic model that they use to associate speakers with quotes based on the content of the quote. Lastly, they created a new corpus that covers the entirety of the novel *Pride & Prejudice*, which was only partially annotated in E&M's work. While their approach appeared to outperform E&M's, their system was very slow, so they did not provide a complete comparison of their system with the system from E&M.

These approaches are summarised in Table 3.1. The table shows that there has been a range of studies on quote attribution that differ significantly in their methods, experimental setup, and results. In particular we note that differing test sets means that there is a clear lack of robust comparison between the work that has been reported on thus far, so

Corpus	Domain	Documents	Tokens	Quotes
SMHC	News	965	601k	7,991
PARC	News	2,280	1,139k	10,526
LIT	Literature	11	407k	5,902
P&P	Literature	1	145k	1,260

Table 3.2: The relative size of the four corpora in terms of documents, tokens, and quotes. Note that PARC and LIT are not fully annotated, so per-document averages will not be accurate.

it is difficult to judge the relative merits of the proposed approaches. In the next section, we will introduce the four corpora that we use for training and evaluation. By presenting results on these four corpora we are able to compare our work with some of the previous studies in the field.

3.2 Corpora

We train and evaluate our methods on four corpora, with two corpora from the news genre and two corpora from the literature genre. Table 3.2 shows the relative size of the four corpora.

3.2.1 Sydney Morning Herald Corpus (SMHC)

Our first corpus is the SMHC, which is a contribution of this thesis. The SMHC was first described in two publications that were completed as part of this thesis. The first was O'Keefe et al. (2012), when it included only direct quotes, and the second was Pareti et al. (2013), when indirect and mixed quotes were added. The corpus contains 965 news articles from the 2009 Sydney Morning Herald.[1] These documents had already been annotated with NEs, pronouns, and coreference information as part of a related research project (Hachey et al., 2013). Common nominal references were not annotated and so entities referenced only via a common noun are not listed as speakers in this corpus (see Examples 11a nad 11b).

[1] http://www.smh.com.au

The corpus was annotated in two main stages. In the first stage, annotators were required to select the NE representing the speaker of each direct quote in each document. The direct quotes had been extracted automatically, so the annotators were also given the option to remove marked spans that were not quotes. These annotations were performed by sixteen annotators, with eleven employed via the outsourcing website Freelancer,[2] while the remaining five were colleagues of the author. From the full set of documents, 400 were fully annotated twice, so that the inter-annotator agreement could be calculated. Raw agreement on the speaker of each quote within these documents was very high at 98.3%.

In the second stage of annotation, a single annotator added indirect quotes and mixed quotes. For each of these quotes, as well as the direct quotes, the annotator selected a single coreference chain, or null, as the speaker of each quote. Lastly, for each quote the annotator marked the *source*, i.e. the specific span of text that references the speaker, provided that the source is within the same sentence as the quote. The source annotations are important, as there are some cases where quotes are attributed to unnamed speakers that have no coreference chain in our gold data, such as Examples 11a and 11b. For these cases the speaker annotation would be null, while the source would link the quote to the text referencing the unnamed speaker. In this chapter, we will ignore quotes that have no annotated speaker, but that have a source, as the correct speaker cannot be predicted with the given candidates. In the next chapter we will propose methods that address this issue.

(11) a. ["There were a few relationships blossoming in the bar,"]$_{quo}$ [said]$_{cue}$ one of her friends.

b. ["The locals and everyone have been great,"]$_{quo}$ [said]$_{cue}$ another.

The annotation guide for the second stage of annotation is included as Appendix A. It was modelled on the guide used in Pareti (2012), so that the SMHC could be comparable to PARC. The SMHC does not include annotations of cue verbs, nested quotes, other attri-

[2] http://www.freelancer.com

bution types, or any of the extra information that is included in PARC that is not listed in this section.

3.2.2 Penn Discourse Treebank Attribution Corpus Extension (PARC)

Our next corpus is an extension to the attribution annotations found in the PDTB (Prasad et al., 2008), which is in turn an extension of the PTB (Marcus et al., 1993), which covers Wall Street Journal[3] articles from 1989. The original PDTB contains several forms of discourse, including assertions, beliefs, facts, and eventualities. While these annotations cover the discourse within the corpus, they do not include all of the attributable text, and so the PDTB cannot be directly used to test quote attribution. However, recent work by Pareti (2012) conducted further annotation and semi-automatically extended the existing annotations to include more information, and also reconstructed attributable text that was only partially annotated.

The annotations are made up of three main spans of text, as well as several other pieces of information that we do not use in this work:

Content: represents the text that is attributable to the speaker.

Cue: the text introducing the quote, which is most often a speech verb.

Source: the text referencing the speaker, which can be made up of NEs, common nouns, pronouns, or can be implicit.

While the content annotations are the same as the SMHC quote annotations, the source in PARC is not directly equivalent to the SMHC speaker annotations. The key difference is that the PARC source annotations are to spans of text, while the SMHC speaker annotations are to coreference chains. This means that for SMHC if a quote is attributed to a pronoun, the canonical form of the speaker is known, while for PARC the pronoun itself would be annotated as the source, with no further information. Furthermore, as the PARC is intended to be a corpus of ARs, rather than quotes, it does not contain any annotations of candidate speakers aside from the source spans that actually have a quote attributed to them.

[3] http://www.wsj.com

To get around this issue we use the BBN Technologies (BBN) Pronoun Coreference and Entity Type Corpus (Weischedel and Brunstein, 2005). This corpus provides a set of gold-standard NEs, common nouns, and pronouns, which can be used as candidate speakers for PARC. The source annotations within PARC can then be aligned with the candidate speakers from the BBN corpus, with the first overlapping entity being set as the speaker. For PARC sources that have no corresponding entity from the BBN corpus we include the source directly as a candidate speaker. While the BBN corpus includes coreference information between the pronouns and their antecedents, it does not include full coreference chains between the NEs and common nouns. As such we use coreference information where it is available, and otherwise put each entity into its own chain. The quotes from Pareti's annotations that have an implicit source cannot be automatically linked to any entity, so they were ignored for the purposes of quote attribution.

In total, PARC contains 10,526 quotes from 2,280 documents. As the corpus was built semi-automatically it is not yet fully annotated, in that there are quotes within PARC that are not marked and have no associated source information. Using a small set of documents that are fully labelled, Pareti estimates that 30-50% of the quotes within the corpus are yet to be annotated. We noted in the previous chapter that this is a serious issue for quote extraction, as there will be positive data that is negatively labelled. For quote attribution this is much less important as the quotes that are not labelled can be ignored, and thus are not included in the training process, either positively or negatively.

3.2.3 Columbia Quoted Speech Attribution Corpus (LIT)

The LIT corpus was developed by E&M for their study into quote attribution. The corpus is a set of eighteen excerpts from eleven works of 19th century literature by six well-known authors. The corpus only covers direct quotes, as indirect and mixed quotes are relatively rare in fiction. The corpus is not fully annotated (see below), as the authors opted for diversity of data, rather than completeness.

This corpus contains NE and common nominal candidate speakers. The NEs were identified using the Stanford NE tagger. The common noun candidates were identified using a regular expression that searches for common nouns that match one of several

Training set		Test set
austen_emma_1	dickens_xmas_1	flaubert_bovary_2
austen_emma_2	dickens_xmas_2	twain_sawyer_1
austen_emma_3	doyle_boscombe	twain_sawyer_2
chekhov_lady	doyle_identity	
chekhov_monk	doyle_league	
chekhov_steppe	doyle_scandal	
flaubert_bovary_1		

Table 3.3: The pieces of documents that are in the training and test sets for the LIT corpus.

patterns with a determiner, an optional modifier, and a head noun. The authors attempted to use a coreference resolution system to link the NEs and common nouns into coreference chains, but found that the coreference systems that are trained on news text were too inaccurate. Instead they used their own system to link NEs with similar names, while common nouns were left as singletons. They do not include pronouns as candidates, as they consider the problem of finding the pronoun's antecedent to be the task of the quote attribution system.

The speaker of each quote was annotated using Amazon's Mechanical Turk[4], with three annotations per quote. In this corpus the annotations link quotes to an individual mention, rather than to a coreference chain. The annotations originally covered 3,578 quotes, however the authors discarded cases where all three annotators chose a different speaker, or where the annotators agreed that the text was not dialogue. While this helps to ensure quality, it is problematic, as cases of three-way disagreement are potentially difficult and should be included in any evaluation after some form of adjudication. Furthermore it results in some sections of the document having gaps, where some genuine quotes are not labelled, which can be a problem for sequence labelling techniques and sequence aware features.

In order to correct this issue we employed two postgraduate students to label the missing quotes, such that the annotated portions of each document form a contiguous block.

[4] http://www.mturk.com

This annotation covered an additional 654 quotes, with a raw agreement score of 80%, calculated from 48 doubly-annotated quotes. While conducting this additional annotation, our annotators noticed that some quotes had incorrect labels, with the speaker closest to the quote chosen. This is a well-known problem with using crowd-sourcing sites, as workers will often choose the first option presented to them, regardless of correctness. To check this issue, one of our annotators checked 400 existing annotations for correctness, and found that 92.5% of the quotes were correctly annotated.

The training and test splits for this corpus are shown in Table 3.3. Note that the test set contains text from works that have been seen before (`flaubert_bovary_2`) as well as works that are by an author who is completely unseen (`twain_sawyer_1` and `twain_sawyer_2`).

3.2.4 Pride and Prejudice Corpus (P&P)

The P&P corpus was developed by He et al. (2013). It covers the entirety of *Pride & Prejudice* by Jane Austen, which is also partially annotated in the LIT corpus. The corpus contains 1,260 direct quotes, which were annotated by an English major. He et al. used the Stanford NER system to identify candidate speakers, which were then manually grouped into coreference chains. This grouping was done by constructing a list of aliases for each character, which is then used to link each instance of each alias to the appropriate chain. While this removes the need to consider the context of each use of the alias, we found that it resulted in some errors. Most notably family members who share a title were often confused, particularly the five characters who can be validly referred to as Miss Bennet. We note that there are few instances of this problem, so we use the corpus as is.

In this corpus the annotations link each quote to a character's coreference chain, rather than to a textually grounded mention of the character. While this is beneficial for some of our later experiments, it does introduce a further consideration, in that unnamed characters are not valid candidate speakers in this corpus. This means that quotes that are spoken by unnamed characters do not have an associated speaker, and so are ignored (shown in Example 12a). Again we note that this occurs in relatively few cases (51 quotes), so we simply follow He et al. and ignore these quotes in evaluation.

(12) a. They ran through the vestibule into the breakfast room; from thence to the library; – their father was in neither; and they were on the point of seeking him up stairs with their mother, when they were met by the butler, who said, [“If you are looking for my master, ma'am, he is walking towards the little copse.”]$_{quo}$

He et al. consider multiple quotes within a paragraph to be a single quote, as part of a one-speaker-per-paragraph assumption. This differs from our definition, where we consider each fragment to be a separate quote that can be merged later. Rather than evaluating our methods on this corpus using our definition, which would make our results incomparable, we evaluate on only the first quote in each paragraph, which is akin to having a one-quote-per-paragraph assumption. During training and classification our system still generates features for the quotes according to our definition, so it is only the evaluation that is affected by this change.

For the training and test sets we follow He et al. and use chapters 19-26 as the test set, and the remainder of the novel as the training set. They used a small amount of data for their development set, however we include this as part of our training data, as our methods were developed on other corpora.

3.2.5 Corpus Comparisons

There are two main factors that influence the difficulty of attributing quotes in the four corpora. The first is the genre. Most of the preprocessing steps, such as POS tagging, NE tagging, parsing, and so on, were developed for news text, and so should perform better on the SMHC and PARC, than they would on LIT and P&P. In addition, news providers typically have a style guide and editorial process that ensures clarity, consistency, and simplicity, which should make quote attribution easier. By comparison, good prose in literature can have these features, but it is not necessary, and in some cases authors will deliberately introduce difficult cases, such as in the following example:

[“A Mrs. Bennet, a Miss Bennet, a Miss Bennet and a Miss Bennet, sir.”]$_{quo}$

For these reasons we consider news to be an easier genre for quote attribution than literature.

	Proportion (%)				Accuracy (%)			
	SMHC	PARC	LIT	P&P	SMHC	PARC	LIT	P&P
Quote-Said-Person	4.0	14.1	16.2	11.0	94.8	99.3	99.8	99.4
Quote-Person-Said	12.8	13.5	1.5	0.0	98.5	94.2	97.3	-
Person-Said-Quote	15.2	22.8	0.0	0.0	90.7	93.0	100.0	0.0
Said-Person-Quote	0.4	0.7	0.0	0.0	15.0	50.0	-	100.0
Quote-Said-Pronoun	0.0	0.0	2.1	7.0	50.0	-	-	-
Quote-Pronoun-Said	11.5	9.1	4.8	2.2	95.8	98.6	-	-
Other Anaphors	3.9	3.7	0.2	1.1	91.9	95.1	-	-
Added	20.5	21.4	27.1	25.4	-	-	-	-
Backoff	24.6	14.6	15.4	15.9	-	-	-	-
Alone	4.2	0.1	11.8	14.8	-	-	-	-
Conversation	2.7	0.0	20.9	22.5	-	-	-	-

Table 3.4: The proportion of quotes in each category and the accuracy of the speaker prediction based on the category. The speaker predictions for pronouns are only available for the SMHC and PARC, as the other two corpora do not have gold standard pronouns available.

The second factor affecting the difficulty of each corpus is the distribution of different styles of attribution. In their work, Elson and McKeown (2010) define a set of categories of quotes, which can be used to gauge how difficult a corpus will be. We adopt their categories, albeit with slightly different definitions:

Trigram – the two non-punctuation elements preceding or following the quote are a mention of an entity and a speech verb;

Anaphor – same as above, except that the mention is a pronoun;

Added – the quote is in the same paragraph as another quote that precedes it;

Conversation – the quote appears in a paragraph on its own, and the two paragraphs preceding the current paragraph each contain a single quote;

Alone – the quote is in a paragraph on its own;

Backoff – the quote matches none of the preceding categories.

The key difference between our definition and theirs is that we allow the *Trigram* and *Anaphor* categories to ignore punctuation, while disallowing cases where the mention and speech verb appear on either side of the quote. We also altered the definition of the *Conversation* category so that it does not rely on the sequence of previous speakers, which would not be available without gold-standard data.

The proportion of quotes in each category in the training sets is shown in Table 3.4. Unsurprisingly, SMHC and PARC have a fairly similar distribution of categories. The largest difference that impacts the difficulty of the corpus is that SMHC uses pronouns more often than PARC, which should make the SMHC slightly harder, as the antecedent of each pronoun would need to be correctly identified. Interestingly, in PARC quotes are more often attributed to named speakers with the speech verb appearing before the speaker, as in the following example:

> [“We have no useful information on whether users are at risk,”]$_{quo}$ [said]$_{cue}$ [James A. Talcott]$_{spk}$ of Boston's Dana-Farber Cancer Institute.

whereas in SMHC the speaker usually appears before the verb:

> [“It's a possibility, it's definitely on the agenda, and we're very serious about making it happen,”]$_{quo}$ [Buckley]$_{spk}$ [said]$_{cue}$.

We speculate that this difference is a result of differing style guides between the SMH and the WSJ. We asked for access to the SMH style guide, but were not given access, so we are unable to confirm this speculation. Despite the large number of quotes affected by this difference, it should have no impact on the relative difficulty of the two corpora, as both structures clearly indicate who the speaker is.

Of the two literature corpora, the P&P corpus appears to be more difficult, as it has more quotes in the *Conversation*, *Backoff*, and *Alone* categories, and fewer in the *Trigram* categories. *Conversation*, *Backoff*, and *Alone* are comparatively difficult, as they do not explicitly indicate the speaker with a speech verb, whereas *Trigram* does.

Table 3.4 also shows that the cross-domain differences are much larger than the in-domain differences. Most notably, the two news corpora have comparatively few quotes in the *Alone* and *Conversation* categories, as they tend not to include monologues or dialogues. This is made up for by the news corpora having more quotes in the *Trigram* and

Backoff categories, however. Overall we expect that quote attribution should be more accurate on news text, as there are more quotes in easier categories, and since journalists generally conform to a prescribed format.

3.3 Experimental Setup

In this section, we will start by describing the task more precisely, and then we will move on to describe the preprocessing, baselines, encoding, and other details required to reproduce our work.

3.3.1 Task Definition

As we pointed out in Sections 3.1 and 3.2, there is not yet a clear definition of quote attribution that researchers have agreed upon. Some definitions consider the task to be to identify the text span that corresponds to the speaker, while others require further processing and disambiguation. The key difference in these definitions is in whether the goal should be to find a speaker that is a simple text span, a coreference chain, or an entity that has been disambiguated back to a knowledge base. In this subsection we will compare these definitions, and note which of the corpora conform to each. We illustrate the differences with the following running example:

> The Defence Secretary, Nick Warner, dismissed reports his department secretly investigated Mr Fitzgibbon, or held concerns that Ms Liu may have had links with China's military intelligence agency. [“We have found not a skerrick of evidence that there is any truth to these allegations,”]$_{quo}$ [he]$_{spk}$ [said]$_{cue}$.

While the four corpora were constructed with different assumptions about what it means to be the speaker, in this chapter we will present results that consider an attribution to be from a quote to a coreference chain. In cases where no coreference information is available, we will fall back to finding the correct text span. We chose this definition as it allows us to present results on all four corpora, in a form that is most comparable with existing research. The drawback of this approach is that all the corpora except LIT use some gold standard information in either the candidate speakers or coreference. This means that the results presented in this chapter are not completely realistic. In the next chapter we will investigate the effect of coreference resolution on quote attribution, and present results that use no gold-standard information.

Text as Speaker

The simplest definition of quote attribution is that the task is to find the source of the quote, i.e. the span of text that refers to the speaker. In the example given, the quote is: "We have found not a skerrick of evidence that there is any truth to these allegations". For this example we would simply need to return the text span he, as it is the source of the quote. This definition would produce output that is most useful for aiding other Natural Language Processing (NLP) tasks that operate solely in-document, such as coreference resolution. However, this definition has two main problems. The first is that not all quotes are explicitly attributed to a speaker, as in the following example, which appears in a paragraph on its own:

> ["There have been lots of papers done, lots of research done, but there are so many issues to get through. We're looking at a range of ways to introduce it, and it's probably a bit premature to say it will be up and running this year, although we aren't ruling that out."]$_{quo}$

In the example there is no text span in the paragraph that the quote could be linked to, and so in some work, such as Pareti (2012), the quote would simply be marked as having an implicit speaker. The other problem with this definition is that for our goal of opinion mining, it is not terribly useful to know that a text span, such as he, is the source of an opinion. We really need the speaker to be further disambiguated in order to associate an opinion with an entity.

Coreference Chain as Speaker

The next definition takes the speaker disambiguation process a step further, by considering a correct attribution to be between a quote and a coreference chain. For the running example a system would need to return the coreference chain containing the mentions he and Nick Warner. This definition mostly gets around the problem of having quotes with implicit speakers, as the quote can be attributed to the coreference chain representing the speaker, rather than to any specific text span in the document. In a very small number of cases this will be insufficient as the speaker will not be mentioned at all in the document, however these cases are rare.

A hybrid form of this definition is what we are using for PARC and LIT, as neither corpus has fully disambiguated entities. In LIT, E&M identified the named entity and common nominal candidate speakers, and ran automated coreference resolution over the named entities. This means that for LIT, a correct attribution links a quote to a coreference chain, with the caveat that the nominal mentions occur as singleton chains, even when they are coreferent with each other or a chain of named entities. For PARC, the original author (who was concerned with a different problem) only annotated a text span for each quote. In our work, we use the BBN entities where possible, as they do have coreference information, while the remaining text spans are treated as singleton coreference chains.

Disambiguated Entity as Speaker

The definition that we use for our own corpus (SMHC) is that, where possible, a speaker should be disambiguated back to a knowledge base. Where it is not possible the speaker should be linked to a coreference chain. In the running example, this means that the marked quote is attributed to a knowledge base entry representing [Nick Warner]$_{spk}$. We chose this definition as it best supports our goal of finding the opinions that entities hold on certain topics. If we were only able to link a quote to a coreference chain or text span, we would face ambiguity when aggregating opinions from multiple documents that are all held by the same entity. A similar definition was used in the construction of P&P, where the knowledge base is a constrained list of characters that are known to appear within the novel.

While we consider a correct attribution to be between a quote and a disambiguated entity, in practice our methods make predictions between quotes and a textual mention of an entity. The reason for this is that it is much easier to write features that consider a quote and candidate pair when the candidate is textually grounded. When evaluating, however, we do not distinguish between separate mentions of an entity, with any mention considered correct if it is disambiguated to the correct knowledge base entry. In Chapter 4 we will discuss this version of the task in more detail, and provide results on the SMHC.

3.3.2 Learning Algorithms

We use two machine learning packages in our experiments. For binary classification we use a maximum entropy classifier from scikit-learn[5] (Pedregosa et al., 2011) with L_1 regularisation, a C value of 0.5, a tolerance of 10^{-6}, and default values otherwise. We experimented with other classifiers and regularisation settings, but found that there was little difference in the results on the development set of the SMHC. For our CRF experiments we use CRFSuite[6] (Okazaki, 2007) with L_1 regularisation and a minimum feature frequency of 3, with other settings left at default. We similarly experimented with other settings, but found that they hurt performance.

3.3.3 Rule-based Baseline

As rule-based approaches are common in the literature (see Table 3.1), we establish a rule-based baseline for this task. The baseline first looks for a speech verb, and then returns the mention nearest the speech verb, or if there is no speech verb it returns the mention nearest the quote. The intuition is that if there is a speech verb in the same sentence as the quote then it is most likely the cue for the quote, and the entity nearest the cue will most likely be the speaker. If there is no speech verb then it is likely that the speaker was mentioned somewhere near the quote, and so taking the nearest is a reasonable fall back.

More precisely the baseline proceeds with the following steps:

1. Search for a speech verb in the text between the end of the quote and the end of the sentence.

2. Search for a speech verb in the text between the start of the quote and the start of the sentence.

3. If a speech verb was found in steps 1 or 2 then return the mention nearest to the speech verb, otherwise return the mention nearest to the quote. In either case the mention cannot be within the quote and the mention must be in the paragraph the quote is in or any preceding it.

[5] http://scikit-learn.org/

[6] http://www.chokkan.org/software/crfsuite/

Aside from being intuitive, this baseline is reasonable as it is able to make accurate predictions for quotes that fall into the simpler categories defined by E&M. Most notably this method should be correct for quotes that match the *Trigram* pattern, and with accurate coreference this method should also work for the *Anaphora* cases.

3.3.4 Text Representation

Our preprocessing and text encoding is similar to the method used in E&M, however with some small changes. The major steps are:

1. POS tag using C&C tools (Curran and Clark, 2003) and dependency parse using the Stanford parser (Klein and Manning, 2002).
2. Normalise all quotes to a single symbol, such that the whole quote is represented in the text as a single token.
3. Normalise all mentions to a single symbol, similarly to the quotes.
4. Normalise all speech verbs to a single symbol, similarly to the quotes. We use a small set of speech verbs that we manually compiled with reference to the SMHC development set.

The key difference between our approach and that of E&M is that we do not exclude tokens, sentences, and paragraphs that contain content that is not relevant to quote attribution. We found that retaining these items improved performance on the various development sets, so we opted to keep them. All of the features that we use in our work are calculated with respect to this text encoding. So, for example, word distance is calculated as the distance in the encoded text, rather than the distance in the original text.

3.3.5 Candidate Generation

When attempting to find the speaker of each quote, it is necessary to generate candidates that the method can assess to see if they are the speaker. There are three main types of candidates:

Proper nouns or named entities, such as John Howard

Corpus	Missing (%)	
	E&M	Ours
SMHC	1.0	0.6
PARC	0.3	0.0
LIT	8.9	2.6
P&P	4.9	2.5

Table 3.5: Percent of quotes in the training sets whose speaker is not among the candidates generated.

Pronouns such as he, she, it, etc.

Common nouns , i.e. noun phrases that do not fit in the above two categories, such as the man, a person, etc.

It is possible to attribute a quote to any of these candidate types, as Examples 9a, 10a, and 10b show. The one caveat here is that we exclude candidates that do not have an NE type of either PER or ORG. We made this decision as quotes are rarely attributed to NEs of other types, and we validated this decision with experiments on the development data.

In their work, E&M generate up to 15 candidates for each quote, with the restriction that the candidates must appear in the paragraph the quote is in or any of the 10 paragraphs preceding it. By contrast, He et al. (2013) consider all of the named characters from P&P, which causes them to miss quotes that are attributed to unnamed speakers, such as the butler.

In experiments on the development set, we found that the He et al. approach of considering all named entities actually hurt performance, as the quote attribution systems had too many candidates to choose from. Furthermore, including all named entities is impractical for some of the texts in the LIT corpus, where many more characters are named than in P&P. More generally, this is a problem for large texts that may refer to a large number of entities.

We also found that the E&M approach was somewhat unsatisfactory for LIT, as there were many cases where speakers were mentioned after they had spoken. As such, we opted to consider 6 candidates before each quote, and 6 candidates after each quote.

Corpus	Sequence Features		
	Gold	Pred	None
SMHC	80	77	78
PARC	92	91	93
LIT	75	49	50
P&P	77	51	61

Table 3.6: Accuracy (%) of our E&M reimplementation with gold standard, predicted, and no sequence features.

We also experimented with restricting the type of mention (i.e. named, pronominal, or common nominal), as well as with restricting the candidates such that each coreference chain would only be included once, however we found that both of these restrictions were slightly detrimental.

We performed a small experiment over the training sets of the four corpora, so that we could test the effectiveness of our candidate generation approach. In the experiment we counted the number of quotes where the speaker was not among the candidates generated, and, using this setup, compared our method to the E&M method. Table 3.5 shows that our method generates the correct speaker more often than E&M over all four corpora. This result is important as the proportion of missing speakers puts a ceiling on the accuracy of any quote attribution system.

3.3.6 Elson and McKeown Reimplementation

As part of our early experiments we reimplemented the core method of E&M, so that we could examine how well their method performed in new genres and how well it would perform without the features that depend on gold-standard information. In order to be as comparable as possible we reimplemented their core method, although we did not replicate all of their experiments. In this subsection we will briefly cover their method and our reimplementation of it.

For each quote the E&M approach considers up to fifteen candidate speakers that appear in the paragraph the quote is in, or any of the ten paragraphs preceding it. In order to choose one of the candidate speakers they train a binary model that is able to predict the probability that each candidate is the *speaker*. This binary classification will result in one separate and independent prediction for each candidate, so they introduce a step where these decisions are decoded. While they experimented with several reconciliation methods, we found that there was little difference between the methods, so we take the candidate with the highest *speaker* probability.

The problem with E&M's method is that they use a sequence feature that relies on gold-standard information about previous decisions. More precisely, they have a feature which is the proportion of quotes in the last 10 paragraphs that were spoken by the candidate. This proportion relies on knowing who said all of the quotes in the last 10 paragraphs, and it is apparent that their work uses gold-standard information for this feature. We reimplemented their method, with the ability to use this sequence feature with gold-standard information, predictions made by the classifier, or with no sequence features at all.

To verify that we had correctly reimplemented their method, we conducted some experiments over the original unaugmented version of the LIT corpus. While most of the experiments in this chapter have a single training and test set, E&M used 10-fold cross validation, so for these experiments we run cross validation. The best result that E&M achieved on this corpus was 83%, whereas our result using 10-fold cross-validation was 82.3%. Further to this, our per-category results were close to the results that E&M found, which indicates that our reimplementation is close to their original work. Potential differences in cross-validation folds, ML parameters, and design and experimental setup are the most likely causes of the slight mismatch in accuracy between our reimplementation and their original work.

While our reimplementation successfully reproduced their results, we stress that the reported accuracy is unrealistic as it uses gold-standard information in the features. Table 3.6 shows the results of using our reimplementation over the four corpora with the gold-standard sequence features, predicted sequence features, and with no sequence features. The LIT results are lower than 82.3% as they are over the extended corpus, with a single run using the training and test splits. This ensures that the methods are tested on unseen authors, which neither our 10-fold cross validation experiments nor E&M ensured.

The main point to note from this table is that replacing the gold-standard features with predicted features results in a significant drop in accuracy of 26% for LIT and 16% for P&P.

In most cases there will only be a small number of entities that are participating in speech events, and in the E&M feature set the sequence feature is the only feature that encodes this. As such, when it is removed there is no way of determining which entities are currently salient. Replacing the feature with a version that relies on previous predictions does even worse. This is caused by a difference between what the feature encodes during training and what it encodes during test time. When the model is training, the sequence feature will have the correct answers for all of the previous decisions, and will receive weight based on this perfect information. During test time, however, the information is imperfect, and can become extremely misleading if the model makes several incorrect predictions in a row. Any features or methods that could encode the same information as the gold-standard sequence features would offer potentially large gains in accuracy.

3.4 Features

In this work we expand on the feature set from E&M with the addition of several new feature types, as well as more features that better capture the information the E&M features were addressing. All of the features are calculated with respect to a given quote-candidate pair, where q represents the target quote whose speaker we want to find, and c represents a given candidate speaker. It is important to clarify here that although we consider an attribution to be between a quote and a coreference chain, we generate features between a quote and a specific mention, so c represents a single mention rather than an entire coreference chain. The reason for this is that it is simpler to build features that are between a quote and a mention. In the sections below, all count features are real-valued unless noted otherwise.

We have divided the features into two sets of features: the standard feature set, which does not rely on previous decisions; and the sequence feature set, which does rely on previous decisions. The features in the standard set are completely unaffected by the decoding strategies we describe in later sections, while the sequence features are dependent on them. In the descriptions below, all features are novel, unless noted otherwise.

3.4.1 Standard Feature Set

The standard feature set covers all of the features that do not rely on previous decisions. As such, they are unaffected by the sequence decoding strategies discussed later in this chapter.

Distance Features

The first set of features measure the distance between q and c, and are intended to give the learner the ability to discriminate between close and far candidates. These features are a reimplementation of some of E&M's features, namely counts of the number of words, sentences, paragraphs, and punctuation marks between q and c, where each count is a separate feature. The features for punctuation marks are counted separately for each type of punctuation.

Paragraph Features

The paragraph features are calculated separately for each of the the two paragraphs preceding q, each of the two paragraphs following q, as well as the paragraph q is in. The features that are calculated for each of the paragraphs are an indicator for whether the paragraph contains c, the proportion of the paragraph's mentions that are part of c's coreference chain, and the proportion of mentions that are part of other chains. We also reimplemented features from E&M, namely counts of the number of words, names, and quotes in each paragraph.

Sentence Features

The sentence features are calculated separately for the two sentences preceding q, the two sentences following q, and the sentence that both q and c appear in, if such a sentence exists. The main intention of these features is to provide extra discriminating information for situations where the learner needs to choose between several candidates that appear in the same sentence as the quote, which is a frequent occurrence. These features also indicate which speakers have been mentioned frequently in nearby sentences.

The features include indicators for whether q and c appear in the same sentence, whether they appear in the same sentence and there are no other candidates in the sentence, whether there is a single speech verb in the sentence, and whether there is a single speech verb in the sentence with c. The remaining features are the distance between the single speech verb (if there is just one) and c, the proportion of mentions of c's coreference chain in the sentence, and the proportion of mentions of other candidates in the sentence.

Context Features

The context features are the normalised form of the five tokens to the left and five tokens to the right of both q and c. The normalisation anonymises names, converts quote marks to a single symbol, converts each token to lower-case, and converts numbers to a uniform symbol. The tokens are indexed by their position relative to either q or c. Each feature is then the index concatenated with the representation of the token, which will indicate whether the token is a speech verb, a mention, a quote, or otherwise will simply be its normalised form. If the context of c contains q or the context of q contains c, then a specific symbol will be used as the value rather than the representation of the token. This allows the features to encode the position of q and c relative to each other. These features are intended to give the learner the ability to discriminate between candidates based on the context that they appear in. The context features are based on features from E&M, which are indicator features for whether tokens near q or c are speech verbs, punctuation marks, or other candidates.

Quote Features

E&M had an indicator feature for whether characters were present or absent within q itself. Our quote features extend this idea. When speaking it is very uncommon for someone to refer to themselves with anything other than a first person pronoun, so these features allow the learner to discriminate against candidates that are mentioned in non-first person form within the quote. The quote features are the proportion of first person and non-first person mentions of c's coreference chain, and of other coreference chains within the quote. These features are calculated for q itself, as well as the two quotes preceding q and the two quotes following q.

Candidate Features

The candidate features are calculated with respect to c and its coreference chain. They are intended to give the learner information about how important c's coreference chain is to the document, and to allow the learner to distinguish between types of mentions, as common noun speakers are much less frequent than speakers that have been introduced by name. These features are the proportion of mentions of c's coreference chain in the whole document, the NE type of c (i.e. PER or ORG), and the nominal type of c (i.e. proper, pronominal, common). E&M had a feature for the raw count of appearances of c in the document, though we found this performed worse than the proportions we propose.

The SMHC also has two apparent stylistic conventions that we include features for. The first is that it is very rare to see a quote before the speaker has been introduced, unless the speaker is introduced in the same sentence as the quote. The second is that the SMHC frequently has mentions of journalists in the first two sentences. We include indicator features to help the learner account for these situations. The indicators are whether c appears within the first two sentences, whether c appears before or after q, and whether c's coreference chain has any mentions before q.

Category Features

In their work, E&M use the category prediction whenever they are available. In this work we do not use those predictions directly, but instead encode them into features. This allows the learner to weight the various category predictions, and enables it to disregard the category predictions if the other features suggest there is a better choice. The actual features are an indicator for whether c is the speaker predicted by q's category, and another indicator for that feature concatenated with the category label. These allow us to have a weight over all the category predictions, as well as a weight for each individual category.

Dependency Features

These features are calculated from the Stanford dependency parse tree (Klein and Manning, 2002) for the sentence that both c and q appear in, if there is such a sentence. The motivation behind these features is to help disambiguate cases where there is an attribution to a candidate, but where other candidates appear between the candidate and the speech verb, as in the following example:

> [Adam Smallhorn]$_{spk}$, 18, from [St Ignatius' College]$_{cand}$, [said]$_{cue}$ [students usually prepared and memorised six essays to cover possible alternatives in section three]$_{quo}$, worth 20 marks.

In the example, the speaker should be [Adam Smallhorn]$_{spk}$, but there is another candidate, [St Ignatius' College]$_{cand}$, that appears between the speaker and the speech verb. Purely distance-based features may prefer [St Ignatius' College]$_{cand}$, as it appears closer to the speech verb. Using the dependency parse, however, can yield the syntactic link between the speech verb and the speaker, which can help the model choose [Adam Smallhorn]$_{spk}$ instead of [St Ignatius' College]$_{cand}$.

The first set of dependency features require a speech verb in the same sentence as both q and c. Indicator features are generated for whether any speech verb has a token in c as a child, whether a speech verb has a token in c as a descendant, and whether a speech verb has a token in c and a separate token in q as a child. The second set of dependency features are the dependency type between c and its parent, as well as the number of nodes between c and its closest ancestor speech verb, if there is one.

Pattern Features

The pattern features encode the sequence of mentions and quotes prior to q, such that common patterns that suggest a particular candidate can be detected. They work by building a string that represents the sequence of all of the quotes, mentions, and pronouns in up to five paragraphs prior to and including the paragraph q is in. The string that encodes these items uses the following elements: `V` for speech verbs; `T` for the target quote (q); `Q` for other quotes; `C` for mentions of c's coreference chain, including c itself; and `O` for mentions of other entities.

The resulting string can then be included directly as a feature. In order to find patterns of various lengths we include substrings of both of these main strings. We generate one substring for each of the five paragraphs, with each substring covering all of the paragraphs between itself and q. Consider the following example, where we are generating features for the the quote-candidate pair of [Carlisle]$_{spk}$ (bolded for clarity) and the second marked quote (beginning "Michael is committed..."):

> There has been a book released and [he]$_{cand}$ has set up a foundation to attract children to swimming. A [Michael Phelps]$_{cand}$ video game is also on the way. A staggering 10 restaurant chains offered [Phelps]$_{cand}$ deals after his success in Beijing. [He]$_{cand}$ chose [Subway]$_{cand}$, partly because of the company's commitment to market him overseas.
>
> ["The ability to partner with Mazda in mainland China presents a unique opportunity that is in line with our overall strategy of developing a relevant marketplace for Michael in China,"]$_{quo}$ [Carlisle]$_{spk}$ said. ["Michael is committed to growing the sport of swimming and the fact that an American Olympian is viewed as an effective marketing vehicle in China is another step forward towards that goal."]$_{quo}$

In this example we consider only the two given paragraphs for brevity and clarity, and ignore any preceding content. With this setup, we would generate the following two strings: `OOOQCVT`; and `QCVT`. As noted, these two strings would be included as features, with a prefix indicating the number of paragraphs covered.

In addition to the strings above, we also generate versions of these strings with the pronouns included. The pronouns are encoded by gender with male pronouns marked with `M`, female pronouns marked with `F`, and other (e.g. it) pronouns marked with `I`. For the example above we would generate the following two additional strings: `MOOMOQCVT`; and `QCVT`. These strings are also included, with both a prefix for the number of paragraphs covered and a marker indicating they are inclusive of pronouns.

Source Features

Many of the quotes in our corpus are attributed to their speaker via an explicit source. While this more explicit link tends to make these quotes easier to attribute, there may still be some ambiguity when there are a large number of candidates in the vicinity of the quote. As such, the motivation for these features is to encourage the learner towards choosing the explicit source when it is available.

The trouble with these features is that using the source from the gold standard essentially gives the learner the correct answer, and would not be available in practice. As such, we need to automatically identify the source. We do this by first searching for a speech verb in the same sentence as q. If there are multiple speech verbs we take the speech verb that is the closest ancestor of q, according to the Stanford dependencies, or if there are no ancestor speech verbs we take the nearest speech verb to q. The subject of this speech verb can then be considered the source of q.

If a source is successfully detected then we generate indicator features for whether the source overlaps c and whether the source overlaps a mention in c's coreference chain. Additionally, if the source is a pronoun we generate indicators for whether the source's gender matches c, another for whether the source's gender matches the longest NE in c's coreference chain, and the concatenation of those two indicators. We determine the gender of NEs by matching the first name to U.S. census data, which indicates whether the name is more common for males or females.

Conversation Features

Though they are not common in news text, conversations form a considerable proportion of the quotes in our two literature corpora. As such, we generate four features for conversational quotes, where we determine that a conversation is taking place if the E&M category of q is *Conversation*, or if q is one of the two quotes leading up to a *Conversation* quote. The *Conversation* category requires a quote to appear in a paragraph on its own, with the two paragraphs preceding the quote's paragraph each containing a single quote. When generating these features we assume that there are two alternating speakers and it is c's turn to speak.

For quotes that are part of c's turn, we generate features for the proportion of non-pronoun mentions that are in c's coreference chain and the proportion that are not in c's coreference chain. We also generate these features for the quotes that are not part of c's turn, such that we have a total of four features. The turn is decided by considering every second quote, both before and after q, to be c's turn, with the other quotes spoken by others. If q is spoken by c then we would expect to see a low proportion of mentions from c's coreference chain in the quotes that make up c's turn.

We demonstrate this with an example:

> Tom considered, was about to consent; but he altered his mind:
>
> ["No– no – I reckon it wouldn't hardly do, [Ben]$_{cand}$. You see, [Aunt Polly]$_{cand}$'s awful particular about this fence– right here on the street, you know –but if it was the back fence I wouldn't mind and SHE wouldn't. Yes, she's awful particular about this fence; it's got to be done very careful; I reckon there ain't one boy in a thousand, maybe two thousand, that can do it the way it's got to be done."]$_{quo}$
>
> ["No– is that so? Oh come, now– lemme just try. Only just a little– I'd let YOU, if you was me, [Tom]$_{cand}$."]$_{quo}$
>
> ["[Ben]$_{cand}$, I'd like to, honest injun; but [Aunt Polly]$_{cand}$– well, [Jim]$_{cand}$ wanted to do it, but she wouldn't let him; [Sid]$_{cand}$ wanted to do it, and she wouldn't let [Sid]$_{cand}$. Now don't you see how I'm fixed? If you was to tackle this fence and anything was to happen to it – "]$_{quo}$
>
> ["Oh, shucks, I'll be just as careful. Now lemme try. Say– I'll give you the core of my apple."]$_{quo}$

Suppose we are generating conversation features for the fourth marked quote (beginning "Oh, shucks..."), with the candidate, c, set to Ben. The quotes that are part of c's turn would be "Oh, shucks..." and "No– is that so? ...". For these quotes we need to count the number of mentions that are part of c's coreference chain, which is zero. We also need to count the number of other mentions in these quotes, which is one, meaning for the features for c's turn we would generate zero as the proportion of mentions in c's coreference chain, and one for the proportion of mentions not in c's coreference chain.

Next, we consider the two quotes that do not form part of c's turn, namely the quotes beginning with "No– no – I reckon..." and "Ben, I'd like to...". For these quotes we calculate the same proportions. There are two mentions of Ben in these quotes, and five mentions of other speakers, so the proportion of mentions in c's coreference chain would be $\frac{2}{7}$, while the proportion of mentions not in c's coreference chain would be $\frac{5}{7}$.

3.4.2 Sequence Feature Set

The following two types of features are distinct from the standard feature set in that they require knowledge of the speakers of previous quotes. In their work, E&M used gold standard information about these previous speakers, which would not be available in practice.

In this thesis, we will present results using gold standard previous speakers, predicted speakers, and results that do not use these sequence features.

We also note that the speaker frequency features were the only features that relied on sequence information in both E&M and in the work we published as part of this thesis, so in both publications they are simply referred to as sequence features. In this work we consider sequence features to also include the speaker pattern features, which are introduced below.

Both the speaker frequency and speaker pattern features require a window of the previous decisions. For cases where the window is not explicitly given, such as the greedy decoding, we use all of the speakers in the 10 paragraphs preceding the quote, which is consistent with E&M.

Speaker Frequency Features

These features are intended to encourage the learner to attribute quotes to speakers who have spoken many of the recent quotes. More precisely, the speaker frequency features are the proportion of quotes within the window that were spoken by c's coreference chain, and the proportion of quotes in the window that were not spoken by c's coreference chain. The proportion of quotes spoken by c's coreference chain was a feature in E&M.

Speaker Pattern Features

While the speaker frequency features tell us the number of recent quotes spoken by various candidates, they say nothing about the *pattern* of speakers. In particular, for *Conversation* quotes we would expect the speakers to alternate, but counts of who spoke recent quotes would only tell us that two entities are salient, not which of them should be the speaker. By contrast, the speaker pattern features are intended to encode the pattern of who spoke recent quotes. Features that encode the pattern of speakers was one of the more novel improvements that He et al. (2013) made over the work of E&M.

We encode the speaker pattern similarly to the pattern features introduced earlier. The key difference is that the speaker pattern features encode the sequence of previous decisions that has been made by the classifier, rather than the sequence of entities that appear in the text. This is done by generating a string representation of the speakers of the previous quotes. Each quote is represented by a single character, where the characters are defined as follows:

- Candidates from c's coreference chain are marked as `C`
- Candidates not in c's coreference chain are labelled with an integer representing their ordinal position preceding the quote, with one integer per coreference chain
- Quotes with no speaker are marked with `N`

Again, similarly to the pattern features, we generate substrings of the above string, in order to find smaller patterns. The difference here is that we generate substrings of each length up to the full length of the main string. We demonstrate this with an example:

> ["Once he died, I got to listen and I loved it,"]$_{quo}$ [Ms Southam]$_{cand}$ said.
>
> Among the sequined tuxedo jackets, hats, white gloves and the odd "Beat It" style jacket, were several celebrities walking the red carpet.
>
> [Teddy Riley]$_{cand}$, who co-produced [Jackson's]$_{cand}$ 1991 album Dangerous, described [Jackson]$_{cand}$ as ["the greatest"]$_{quo}$.
>
> ["Every day I spent with him I had fun, but...working on the record was a lot of hard work, because he settles for nothing less than great."]$_{quo}$
>
> Australian Idol winner [Wes Carr]$_{cand}$ said [Jackson meant everything to him.]$_{quo}$
> ["He's the reason why I do this as a living now,"]$_{quo}$ [he]$_{spk}$ said.

If we were generating features for the mention [he]$_{spk}$ and the final quote, we would get `C112`, `C11`, `C1`, and `C`. In the first example the `C` refers to the quote starting "Jackson meant...", the `1`s refer to the quotes "Every day..." and "the greatest", and the `2` refers to "Once he died...", while the other strings are simply substrings of the first example.

We also generate a condensed version of these features, where consecutive quotes by the same speaker are merged, such that the string contains no repeated characters. For the above example these features would be `C12`, `C1`, and `C`.

3.5 Class Models

In most NLP tasks, such as POS tagging or NE tagging, there is a fixed set of labels that a machine learning algorithm can learn from and predict. In quote attribution, however, the set of speakers is unbounded, so it is impractical to have training data for every potential speaker. Instead, we need a class model that anonymises the speakers, such that training data applies more generally. In this section we will introduce two class models that each address this issue. In order to more clearly explain the difference between the various models, we will consider the following running example:

> [The Defence Secretary]$_{cand}$, [Nick Warner]$_{cand}$, dismissed reports [his department]$_{cand}$ secretly investigated [Mr Fitzgibbon]$_{cand}$, or held concerns that [Ms Liu]$_{cand}$ may have had links with [China's military intelligence agency]$_{cand}$. ["We have found not a skerrick of evidence that there is any truth to these allegations,"]$_{quo}$ [he]$_{cand}$ [said]$_{cue}$.

In this example we can take both the quote and the candidates to be given (marked with [ ]$_{quo}$ and [ ]$_{cand}$ respectively), with the correct speaker being the coreference chain that contains [The Defence Secretary]$_{cand}$, [Nick Warner]$_{cand}$, and [he]$_{cand}$.

3.5.1 Binary

The binary class model, which was introduced in E&M, works by labelling the correct candidate as the $speaker$ and the other candidates as $\neg speaker$. During training a set of candidates are generated according to the method outlined in Section 3.3.5, with each quote-candidate pair contributing a separate training instance to the classifier. Once trained, the learned model can then predict the probability of $speaker$ versus $\neg speaker$ for the set of generated candidates. This set of decisions then needs to be decoded to a single prediction of which candidate is the most likely speaker. While E&M experimented with several decoding methods, we chose to take the candidate with the highest $speaker$ probability, as we found little difference between the methods. This class labelling would model the running example as below, where [ ]$_{spk}$ indicates $speaker$ and [ ]$_{\neg spk}$ indicates $\neg speaker$:

> [The Defence Secretary]$_{\neg spk}$, [Nick Warner]$_{\neg spk}$, dismissed reports [his department]$_{\neg spk}$ secretly investigated [Mr Fitzgibbon]$_{\neg spk}$, or held concerns that [Ms Liu]$_{\neg spk}$ may have had links with [China's military intelligence agency]$_{\neg spk}$. ["We have found not a skerrick of evidence that there is any truth to these allegations,"]$_{quo}$ [he]$_{spk}$ [said]$_{cue}$.

The above example shows that each candidate has been labelled as either $speaker$ or $\neg speaker$. There are two key points to note however. Firstly, each candidate will be considered as a separate training or classification instance, rather than the whole set of candidates being considered at once. Secondly, the example shows that although multiple candidate mentions are in the correct coreference chain, only the *source* of the quote is labelled as the $speaker$. We experimented with labelling multiple candidates in the same coreference chain as the $speaker$, however we found that this hurt performance, as positive training instances were associated with mentions that were not introducing speech. It is also worth clarifying here that although we only provide one $speaker$ during training, the learner can predict any of The Defence Secretary, Nick Warner, or he, during test time and it would be considered correct.

The strength of this method is that it learns one decision boundary that separates $speaker$ from $\neg speaker$. This boundary is insensitive to who the candidate is, its position relative to the quote, and any other factor, unless those factors are explicitly encoded into features. The main drawback of this method is that the various candidates are unable to directly compete for probability mass, as they are each classified independently. In order to ameliorate this issue we include all of the features for each of the candidates in every other candidate's feature set. We do this by prefixing the features from candidates not in the current candidate's coreference chain with an 'O', while candidates within the current candidate's coreference chain are prefixed with 'C'. Any features that appear for multiple non-speaker candidates are summed, including indicator features which are treated as though they have a value of one. As an example, suppose that we are generating features for the candidate $[\text{he}]_{spk}$ above. With this scheme we would include the features for $[\text{he}]_{spk}$, plus the features for The Defence Secretary and Nick Warner prefixed with 'C', as well as the features for all the other mentions prefixed with 'O'. This enables some more direct competition between the candidates in the binary class model.

The other drawback of the binary class model is that its output cannot be easily transformed into a sequence, as it makes multiple classifications per quote. As such this method is unsuitable for the CRF experiments, so we do not attempt to train a CRF using this class model.

3.5.2 Positional

The positional class model works by labelling the candidate mentions according to their position relative to the quote. Candidates appearing before the quote are labelled with the prefix pre and a number representing their ordinal position amongst the candidates preceding the quote. Similarly, the candidates that follow the quote are labelled with the prefix $post$ and their ordinal position following the quote. The learner can then learn from and predict these labels directly. The running example would be labelled as below:

> [The Defence Secretary]$_{pre6}$, [Nick Warner]$_{pre5}$, dismissed reports [his department]$_{pre4}$ secretly investigated [Mr Fitzgibbon]$_{pre3}$, or held concerns that [Ms Liu]$_{pre2}$ may have had links with [China's military intelligence agency]$_{pre1}$. ["We have found not a skerrick of evidence that there is any truth to these allegations,"]$_{quo}$ [he]$_{post1}$ [said]$_{cue}$.

A single mention must be chosen as the correct speaker in order to train using this class model. In situations where coreference information is available this can be a problem, as there may be multiple candidates that are in the correct coreference chain. More concretely, from the example above all of [The Defence Secretary]$_{pre6}$, [Nick Warner]$_{pre5}$, and [he]$_{post1}$ could be considered correct. We resolve this in a similar fashion to the binary model, where we choose the candidate that corresponds to the source as the speaker. The source candidate is more likely to be explicitly linked to the quote via a cue verb than any of the other candidates in the correct coreference chain. For the given example we would consider [he]$_{post1}$ to be the correct choice during training. Though this is an issue for training, during classification any of the candidates that are in the correct coreference chain are considered correct, similarly to the binary class model.

The features present a slight challenge for this model, as the features listed in Section 3.4 are all relative to a single quote-candidate pair, whereas this model needs features for all of the candidates to be included in a single instance. We resolve this by generating all of the features for each quote-candidate pair and prefixing them with the class label of the given candidate. In our running example, the instance would have the features for all of the listed candidates, with the features for [The Defence Secretary]$_{pre6}$ being prefixed with $pre6$, the features for [Nick Warner]$_{pre5}$ begin prefixed with $pre5$, and so on.

The positional class model essentially reverses the advantages and drawbacks of the binary class model. On the plus side, it encodes the candidates so that they are classified in one go, subject to the normal one versus rest decoding that maximum entropy learners employ. This means that the candidates can compete directly for probability mass. On the negative side, however, it removes the single $speaker$ versus $\neg speaker$ decision boundary, which means that a separate decision boundary has to be learned for each class label. This dramatically reduces the amount of positive evidence available for each label, with particularly sparse evidence for the higher-numbered labels.

3.6 Sequence Decoding

In Section 3.4.2, we noted that the sequence features could be used with gold standard values, predicted values, or not at all. When using gold standard values, the features that are generated for classification should be the same as the features generated during training, so the model's weights will make sense. If we use predicted values, however, there will be an issue as the gold standard values used during training will be quite different to the predicted values that are generated for classification. Any errors that occur during classification have the potential to be misleading for future decisions, which can result in mistakes compounding. This problem can be ameliorated by introducing a sequence decoding step, which is able to make trade-offs by choosing candidates that are less likely in order to arrive at a more likely overall sequence of decisions.

In this setting the sequence that we want to decode is the sequence of attributions of speakers to quotes. When making a prediction it would be ideal to base it on the entire history of previous decisions. Unfortunately, this is often impractical due to both the time complexity, and the difficulty in reliably estimating probabilities for previous decisions that occurred well before the decision under consideration. As such, we define a history of previous decisions, h, which is available when classifying a given quote.

There are two ways, broadly, that the sequence can be encoded. The first way is through features, such as the sequence features that we defined in Section 3.4. In this method, the probability that a given candidate is the speaker of a given quote can change based on how the features represent the history, h. The alternative approach is to encode the sequence in transition weights, which represent the chance of transitioning from one class label to the next. In this approach, the transition weights encode the probability of transitioning from the sequence of previous decisions, h, to a given candidate label. Both of these approaches require some form of sequence decoding to arrive at the optimal sequence of decisions.

3.6.1 Greedy Decoding

The greedy algorithm is the simplest form of decoding, in that it calculates the probability of each label based on a single fixed history of the previous predictions. To do this the greedy algorithm uses a standard classifier at each step, with h being filled with a single set of previous decisions made by the classifier. The sequence features can then be calculated with respect to h. More concretely, at the first decision point the classifier will make a prediction with no history, as none is possible. At the second decision point the classifier will use the single prediction it made on the first decision point to calculate its sequence features. At the third decision point the classifier will use the decisions made at the first and second decision points to calculate the sequence features, and so on.

This set up is extremely simple, but has the drawback that the greedy algorithm is unable to make any trade-offs between previous decisions and the current prediction. It may be that the classifier could make a better decision about the current attribution if it considered a different history. In effect, the decoder is choosing a local maximum at all times, which can result in it following a relatively low probability path through the sequence. This means that, in general, greedy decoding will not return the most likely sequence of decisions.

As greedy decoding is efficient, we only restrict the number of speakers in h to the quotes that were in the last ten paragraphs, which helps ensure the speakers are salient to the current quote. This is a Markov assumption, with the slightly unusual feature that the number of previous decisions that we consider is not fixed. This restriction also applies to our results that use the gold-standard sequence features.

3.6.2 Viterbi Decoding

When arriving at a particular decision, the greedy algorithm simply picks the most likely candidate at that decision point using the sequence of previous decisions that the classifier had predicted. The trouble with this is that there may be a more likely *sequence* of previous decisions that the greedy algorithm did not consider. In other words, as greedy decoding only considers the conditional probability of a candidate given its features and the previous decisions in h, it will not necessarily find the sequence with the highest joint probability.

To find the optimal sequence of decisions, we first need to be able to find the joint probability of a given sequence. In an unconstrained formulation of the problem, calculating the joint probability of a given sequence requires calculating the probability of every speaker in the sequence, given the full set of decisions prior to that speaker. Unfortunately, this is impractical, particularly for long documents with many quotes, such as P&P. In any case, it is unlikely that the speaker of the last quote in P&P depends much on the speaker of the first quote, or even on the speaker of a quote a few paragraphs prior.

Rather than considering all of the previous decisions, we can instead consider just the decisions within a limited history, h. Similarly to greedy decoding, this is also a Markov assumption, except that we have the more usual situation that the history of previous decisions is of a fixed size. Using this assumption, the probability of a speaker at a given point depends on its standard features, and on the sequence features calculated using a given history, h. We can then calculate the joint probability of a sequence by multiplying together these conditional probabilities.

While the Markov assumption makes calculating the joint probability of an individual sequence tractable, there is still the problem of finding the optimal sequence. An exhaustive search of this space would be impracticably inefficient, as we would have to calculate the joint probability of all possible sequences. However, some of these sequences can be ignored. The reason for this is that there are multiple paths to get to any decision point, but, because of the Markov assumption, only the steps taken within h matter. For these cases, we only need to consider the sequence prior to h that has the highest joint probability, as the other sequences already have lower probability and have no way to impact the current or future decisions, due to the Markov assumption. The Viterbi algorithm exploits this fact, and allows us to find the sequence with the highest joint probability, whilst avoiding these less likely cases. This style of model is a Maximum Entropy Markov Model (MEMM), formulated similarly to the model proposed by Ratnaparkhi (1996) for POS tagging.

The sequences using both the binary class model and the positional class model can be decoded using the Viterbi algorithm, so we present results using both. We also experiment with varying history sizes, in order to gain insight into how the previous decisions impact the current decision. Though the Viterbi algorithm is able to find the best sequence of probabilities without the need for an exhaustive search, it can still take an impractical amount of time to run. As such we employ a beam and ignore all but the ten most promising histories at each decision point. This removes the theoretical guarantee of optimality, although in practice a beam search will yield the optimal or a near-optimal sequence.

3.6.3 Conditional Random Field (CRF) Decoding

The approach described in the previous section allows us to uncover the optimal sequence of decisions according to the model. However the model has a problem, in that during training, each decision – and the sequence it is conditioned on – is considered as a separate training instance, without regard for the rest of the sequence. In effect, this means that each decision is uniformly weighted. This prevents decisions at different points in the sequence from being weighted against each other, which is known as the *label bias* problem (Lafferty et al., 2001).

Instead of treating each decision as a training instance, CRFs treat entire sequences as training instances. This allows them to avoid the label bias problem, as the error in the model can be calculated with respect to entire sequences. This allows the optimisation method to make tradeoffs between decisions. The drawback of this approach is that it requires the forward-backward algorithm during training, which makes CRFs much slower to train. This problem is minor for small history sizes, but becomes problematic for larger

Corpus	Gold seq.	Rule	No seq.	Greedy	Viterbi		
					$h = 1$	$h = 2$	$h = 5$
SMHC	*93*	85	92	92	93	92	92
PARC	*97*	78	97	97	97	97	97
LIT	*70*	55	64	64	63	66	58
P&P	*87*	55	79	84	77	83	78

Table 3.7: Test set accuracy (%) with the binary class model. Italicised results indicate gold standard information is used.

Corpus	Gold seq.	Rule	No seq.	Greedy	Viterbi			CRF
					$h = 1$	$h = 2$	$h = 5$	
SMHC	*89*	85	89	89	89	89	89	90
PARC	*95*	78	95	95	95	95	95	95
LIT	*56*	55	54	54	52	56	53	53
P&P	*61*	55	54	53	50	62	52	61

Table 3.8: Test set accuracy (%) with the positional class model. Italicised results indicate gold standard information is used.

windows. As such, for the CRF experiments in this work, we restrict the sequence features to just the label transitions.

Though the positional class model can be used directly in a CRF, the binary class model cannot be used without significantly reworking the model. The problem is that the CRF depends on learning the transition probabilities between class labels. So while the positional model will have transitions such as pre_1 to $post_1$, the binary model will have a number of separate labels of $speaker$ versus $\neg speaker$ for each decision. As it is clearly meaningless to learn the probability of transitioning from $speaker$ to $\neg speaker$ or $speaker$, we do not include CRF results using the binary class model.

3.7 Results

Tables 3.7 and 3.8 show the results of our quote attribution experiments. The first point that we would like to highlight from these tables is that using machine learning for this task yields much better results than a simple rule-based approach. For the two news corpora, the rule-based system achieved 8% (SMHC) and 19% (PARC) below the best learned method, while on the two literature corpora the rule-based system achieved 11% (LIT) and 29% (P&P) below the best learned results. This tells us that there are aspects of the problem that the rule-based systems have failed to capture, which we have successfully modelled with our machine learning approaches.

If we compare the rule-based versus greedy results for P&P, the main categories with a significant difference in accuracy were *Alone* with 32% versus 86%, *Backoff* with 49% versus 78%, and *Conversation* with 54% versus 81%. While a more syntax-aware baseline might perform better on the *Backoff* cases, it would have no advantage for either *Alone* or *Conversation*, as there is no candidate within the sentence, and thus no syntactic link to exploit. This demonstrates that while syntactic information can benefit quote attribution, purely syntactic methods are in general not sufficient, as there are categories of quotes, such as *Alone* and *Conversation*, where there is no syntactic link between the quote and its speaker.

While our E&M reimplementation shows that there is a large drop in accuracy when gold-standard features are removed or replaced with predicted features, Tables 3.7 and 3.8 show that when the full feature set is used the difference is much smaller. For both of the news corpora the effect of removing the gold-standard sequence features or replacing them with predicted features results in a drop of at most 1%. This indicates that our extra features are successfully encoding much of the information that the gold-standard sequence features encode.

For the two literature corpora there is a larger difference of 6% when using the binary class model on LIT, and 3% when using the binary class model on P&P. Intuitively, it makes sense that the sequence features would play more of a role for the literature corpora, as they have a higher proportion of quotes in the *Conversation* category, which benefit from knowing who the speakers of previous quotes were. These two corpora are also less likely to explicitly indicate speakers once they are in a quote-heavy portion of the text, as the reader will often understand who is speaking based on the content of the quotes. In these cases the pattern of previous speakers can be very helpful.

Somewhat surprisingly, the Viterbi algorithm did not close the gap between using the gold-standard sequence features and using the predicted sequence features. For the news corpora there was no real gap to close, so the Viterbi results are equivalent to the results that use predicted sequence features. However for the two literature corpora, the Viterbi algorithm was outperformed by the greedy decoding for P&P and only achieved a small improvement of 2% over the greedy algorithm for LIT.

These results are somewhat counter-intuitive as the Viterbi algorithm should be finding higher probability sequences of decisions than the greedy algorithm, and should therefore yield better results. However, there are two factors that confound this intuition. The first is that the size of the window, h, is limited in the Viterbi results in order to keep the run time reasonable. This means that the Viterbi results have less information available when generating features and may suffer some loss in accuracy as a result. The other confounding factor is that the Viterbi results still rely on the gold standard features during training, and so may still place too much weight on the sequence features. This effect can potentially compound errors, where the decoding process prefers sequences that seem good to the sequence features, while ignoring the other features. This problem particularly manifests in the classifier predicting long sequences where a single candidate is always chosen, regardless of the context of the quote.

3.7.1 Class Models

Of the two class models, the binary model, shown in Table 3.7, generally performed better than the positional model. While the results for the two news corpora are fairly close, the results on the literature corpora show a large difference between the best performing binary result and the best performing positional result of 10% on LIT and 12% on P&P. This indicates that it is better to have a single decision boundary that learns the difference between $speaker$ and $\neg speaker$, rather than having multiple boundaries that are dependent on the candidate's position relative to the quote.

One of the advantages of the positional model is that it permits training a standard CRF. The results in Table 3.8 show that the CRF results are fairly underwhelming, and are lower than the binary results. Part of the reason for this is that the positional class model splits the positive evidence between speakers and quotes over several class labels. A further issue however, is that the sequence of positional class labels is not necessarily meaningful. In most cases the candidates nearer to the quote will be the speaker, so most of the transitions that the CRF learns will be that each label is more likely to transition to lower numbered labels than to higher numbered ones.

A better class encoding would allow the CRF to have a consistent set of speakers between quotes, so that it could learn patterns such as dialogues and monologues. We experimented with adding two class labels to the positional model that allowed the CRF to choose the same speaker as the previous quote, or the same speaker as the quote before that. These extra class labels add a further difficulty in that they introduce even more conflicting correct answers, with only one answer able to be chosen during training time. We experimented with preferring the previous quote labels over the standard positional labels and preferring the standard positional labels over the previous quote labels. In both cases the CRF performed worse, as in the first case the prevalence of the quote labels meant that few instances were associated with the standard positional cases. When the positional labels were preferred the quote labels were only used when the candidate did not appear among the positional cases, which happens in only a very small number of cases.

3.7.2 PARC Results

The results from PARC clearly stand out from the other results, in that aside from the rule-based approach, most of the methods achieve accuracies that are very close to 100%. The reason for these results is that the corpus was constructed for a slightly different purpose, and so has assumptions that differ from our other corpora. In PARC, quotes are annotated with their source, which is a span of text that references the speaker of the quote. When there is no source, i.e., when a quote is attributed to its speaker implicitly, there is no source annotation. We chose to remove these quotes from evaluation, as these quotes will have been spoken by someone, however there is no way of determining if a predicted speaker is actually correct. Unfortunately these cases tend to be more difficult, so this makes the results on PARC appear better than what is realistic. Furthermore, PARC covers many of the same documents that are used when training upstream NLP components, such as POS taggers and parsers. As such, much of the information that the system relies on will be more accurate than it would be on unseen documents, which will also make the results on PARC appear better than they would realistically be.

3.7.3 Comparisons with Previous Work

One of the problems with prior research into quote attribution is that there have been few comparisons between methods. As such, in this section we will compare our results with the results of other researchers who have used the same data.

Elson and McKeown (2010)

Comparing our results to the E&M results is non-trivial as their experimental set up is quite different to ours. While we nominated fixed training and test sets, they ran 10-fold cross validation, with each fold taking quotes from potentially many documents. This means that they may have used training data from the same document that they were using for test data. Furthermore their features rely on gold-standard information, which would not be available in practice. Nonetheless some broad comparisons are possible.

The best result that they achieved in their work was 83% accuracy, using gold-standard information for some of the features. Our reimplementation of their work achieved 82.3% on the data they had available, which indicates that it is reasonably close to their method. Once we included the extra data, the result from our reimplementation dropped to 75%. This result is better than the best result we achieved on their corpus using our own method with gold-standard features, which achieved only 70%. However when we remove the gold-standard features from our reimplementation of E&M, the results drop to 50%, which is worse than our rule-based method. By comparison, our best method that does not use gold-standard information achieves 66% accuracy. Despite the differences in methodology, it appears that our method performs better.

He et al. (2013)

The most straightforward comparison of our work with He et al. (2013) is on the P&P corpus where we have the same data with the same training and test splits. Using gold-standard information our best result is 86.7%, which is slightly higher than their gold-standard result of 86.5%. Without the gold-standard information our best result is 83.6%, compared to their best of 82.5%. While we achieved slightly higher performance in both cases, the difference is not statistically significant, so both methods appear to be equivalent.

He et al. also present results on two of the documents from the LIT corpus, namely *Emma* and *The Steppe*. Their results use the versions originally presented by E&M, rather than our updated version. To match this, we evaluated our greedy method with the binary class model on the original versions of these two documents, using the remainder of the documents for training. Our greedy model achieved 70.3% and 73.6% for *Emma* and *The Steppe* respectively. This is lower than He et al's best results on these documents, which were 74.8% and 80.3%. As our methods were developed using the SMHC development set, it is not surprising that our results are not as strong on the literature genre.

3.7.4 Incremental Improvements

As part of this thesis we published two works that use early versions of our methods and data (O'Keefe et al., 2012; Pareti et al., 2013). In this section we will note the improvements that we have made since then and compare results.

O'Keefe et al. (2012)

In O'Keefe et al. (2012) our best results across the SMHC, PARC, and LIT, were 92.4%, 84.1%, and 53.3%, respectively. The core difference between these and our current results, is that our current results use a much larger feature set. Our best result on the SMHC is now 92.5%, which is only a marginal improvement. However this result is on a newer version of the corpus that includes indirect and mixed quotes, which makes it a more difficult corpus than what was used for the older results. Our current result using only the direct quotes is 93.6%, which is 1.2% higher than the 92.4% using our earlier methods. On LIT the result is even better, as in O'Keefe et al. the best result was the rule-based system which achieved only 53.3%. This compares very poorly to the 65.8% achieved by our current methods, which represents a gain of 12.5%.

Pareti et al. (2013)

Our results have also improved over the attribution results presented in Pareti et al. (2013), which used the same methods as the O'Keefe et al. (2012) results. The difference between the O'Keefe et al. results and the Pareti et al. results is that the Pareti et al. results use the current versions of both the SMHC and PARC, which allows us to compare results directly. The best result in Pareti et al. on the SMHC was 87%, which is over 5% lower than our current best result of 92.5%. There was an even larger improvement on PARC, which went from 77% to 97%. This biggest factor in this difference was the features that allowed the learner to discriminate against common noun candidates, which rarely act as speakers.

3.8 Summary

In this chapter we have evaluated several approaches to quote attribution on both newswire and literature, and, where possible, we have compared these results to previous work. Our results show that using a simple rule-based system will achieve between 78% and 85% on news text and around 55% on literature. However, this can be strongly outperformed by machine learning approaches, which were able to achieve 93% and 97% on the two news corpora and 66% and 84% on the literature corpora.

We cannot directly compare these results to all of the previous work on this task, however, we have made direct comparisons where possible. Our news results represent the first large-scale evaluation of quote attribution in this genre, and so in our view these results are state-of-the-art. On the literature corpora, we made direct comparisons with two previous publications. The work of Elson and McKeown (2010) used gold-standard features that are not realisable in practice. When we removed these features from our reimplementation of their work, we saw a large drop in accuracy, which meant that our work outperformed theirs. Our comparisons with the work of He et al. (2013) showed that our methods achieved marginally better results on their P&P dataset, however they achieved better results on two of the documents from the LIT corpus.

This chapter presents the results of quote attribution using gold-standard quote spans, and the candidate speakers that come with each corpus. In the next chapter we examine the full pipeline of systems, namely quote extraction, mention detection, coreference resolution, and finally quote attribution. This gives us more realistic results when we consider all of the potential sources of errors.

4 · Pipeline Effects

In our evaluation of quote extraction and attribution methods thus far, we have taken the candidate speakers as given, and in the case of quote attribution we also took the quoted content itself as given. While these assumptions make it simpler to compare our methods with existing work, they are not realistic, as this perfect information would not be available in practice. As such, in this section we will evaluate the full pipeline of candidate extraction, quote extraction, and quote attribution. This is a core contribution of this work, as it gives us realistic results that will tell us how many errors we should expect for our opinion mining work.

One of the key challenges in performing this whole-pipeline evaluation is that the different corpora make different assumptions about what it means to be the speaker of a quote. These differing assumptions make it hard to judge whether an automatically generated speaker correctly aligns with a speaker from the gold standard. In particular, it is important to distinguish between whether we are interested in the span of text representing the speaker, i.e. the source, or whether we are interested in the coreference chain, or even a fully-disambiguated link back to a KB. If we are only interested in finding the source of the quote, then we only need to automatically detect mentions, and we can ignore coreference information. However, for most potential downstream applications, including opinion mining, it is more informative to disambiguate the speaker back to a coreference chain or KB entry.

In addition to the different assumptions about what it means to be the speaker, the corpora also come with different types of mentions and coreference information, which makes it difficult to compare results across corpora. The LIT corpus, includes automatically identified named entities and common nouns, but does not include pronominal references and only includes coreference information between named entities. P&P includes automatically identified named entities with limited gold-standard coreference, but does not include pronouns or common nouns. The SMHC includes gold-standard named entities and pronouns, as well as gold-standard coreference, but does not include common noun candidates. Finally PARC is intended to cover attribution more generally, and so does not include any candidate speakers except for those that have a quote attributed to them. By running automatic mention detection and coreference resolution over the four corpora, we can get more consistent results across the corpora.

In addition to the primary contribution of this chapter, which is to run a full pipeline evaluation, we also evaluate pairs of systems in the pipeline, so that we can understand where errors are occurring. The first pair that we examine is quote extraction with automatic mentions and coreference, so that we can determine to what extent our ability to find quotes is impacted by automatic coreference. Next, we move on to examine quote attribution with automatic mentions and coreference. This is a trickier problem, as the annotations link quotes to gold-standard coreference chains. To solve this, we define two separate alignment methods, which allow us to align the gold-standard speaker annotations with the automatic chains. Next, we evaluate the complete pipeline of coreference resolution, quote extraction, and quote attribution, which gives us realistic results for the full task. Lastly, we add NEL to the pipeline, which allows us to evaluate how accurate the full pipeline is for those quotes that can be linked to a KB entry.

Parts of this Chapter are based on work that was published in the proceedings of the Australian Language Technology Workshop (2013).

4.1 Coreference Resolution

Coreference resolution is the task of grouping mentions, typically noun phrases, into clusters which refer to the same real world entity. These clusters can contain any assortment of pronouns, common nouns, and named entities. The task has largely been attempted by determining whether a particular mention anaphorically links to another mention that precedes it. Both supervised and unsupervised methods have been proposed for coreference resolution, and both have been competitive in recent shared task evaluations. Research into quote attribution has ignored the impact that these different approaches could have. Furthermore, the four large-scale corpora that exist for quote attribution are inconsistent in their treatment of mention detection and coreference resolution, which makes comparisons between the corpora difficult.

By running an automatic coreference resolution system over each corpus, we are able to standardise the mentions and coreference information that is available, which makes results on the four corpora much simpler to compare. As there is a lot of variability between systems, we present results using three coreference systems, which allows us to make more robust comparisons between the quote attribution methods. The three systems were chosen to capture the range of approaches that have been proposed for coreference resolution.

By comparing the effect of different coreference systems on quote attribution, we are also able to extrinsically evaluate the coreference systems. Intrinsic evaluation of coreference resolution is challenging (Kummerfeld and Klein, 2013), as the existing metrics penalise the various types of errors differently. This can make it difficult to evaluate the relative performance of two coreference systems, as one system may do better on one metric, while the other system does better on a second metric. By performing an extrinsic evaluation of coreference resolution, we are able to directly evaluate the effect that various systems have on the downstream task that we are interested in, which in this case is quote extraction and attribution.

In the following two subsections, we will describe the mention types and coreference information that is available for the four corpora, and we will then move on to describe the three coreference systems that we use in this chapter.

4.1.1 Coreference in the Corpora

Table 4.1 shows a summary of the types of mentions that are included as candidate speakers and the coreference information that is available in the four corpora. In terms of candidate speakers, the table shows considerable variance amongst the corpora with no corpus including a full set of gold-standard candidates.

Corpus		SMHC	PARC	LIT	P&P
Documents		965	2,280	11	1
Tokens		601k	1,139k	407k	144k
Quotations		6,705	9,961	3,486	1,692
Entities	Proper	Gold	Gold	Auto	Auto
	Pronouns	Gold	Gold	-	-
	Common	-	Gold	Auto	-
Coref	Proper	Gold	-	Auto	Gold
	Pronouns	Gold	Gold	-	-
	Common	-	-	-	-

Table 4.1: Comparison of the four corpora in terms of size and the candidate speakers included.

Sydney Morning Herald Corpus (SMHC)

The candidate speakers for the SMHC consist of gold-standard annotations of proper nouns and pronouns, which were completed as part of a separate research project (Hachey et al., 2013). Both the proper nouns and the pronouns were manually merged into coreference chains, which were linked to cross-document representations of the real-world entity the chains refer to. Additionally, some entities were linked to a KB entry, which we discuss in more detail in Section 4.5. Annotating a candidate as the speaker of a quote in this corpus involves linking the quote to the whole *coreference chain*. This corpus does not include common noun mentions, and so quotes attributed to entities that are only referenced via common nouns do not have speaker annotations.

In addition to the speaker annotations, the SMHC also has annotations of the source of each quote. The source is the specific span of text that the quote is attributed to, with a restriction for this corpus that the span must be in the same sentence as the quote. The source does not necessarily need to align to one of the mentions, so quotes that are spoken by entities that are only mentioned via common nouns can still have source annotations. In the previous chapter, we evaluated our methods over all quotes that had an annotated speaker, and excluded those that had no speaker annotation. However, in this chapter, the coreference systems are able to generate common noun candidates that can be linked to the quote via the source annotation, so we will include these extra quotes in the evaluations in this chapter.

Penn Discourse Treebank Attribution Relations Corpus (PARC)

Unlike the other corpora, PARC (Pareti, 2012) does not link quotes to coreference chains, and instead marks the source span of each quote. We used the BBN pronoun coreference and entity type corpus (Weischedel and Brunstein, 2005) to add candidate speakers. Using the BBN corpus, we get gold-standard named entities, pronouns, and common nominal references, though the only coreference information is between pronouns and their antecedents.

Before we can use this corpus, we need to align the BBN mention annotations with the PARC source annotations. We do this by finding the first BBN entity that is a subspan of each *source* annotation from PARC, which is then set as the speaker. Where no BBN entity matched, we inserted Pareti's *source* itself as an additional mention, so that the quote attribution methods are able to find a mention for all of the annotated quotes.

Columbia Quoted Speech Attribution Corpus (LIT)

Elson and McKeown (2010) found candidate speakers in LIT by identifying proper nouns with the Stanford NE tagger and common nouns through patterns that have a combination of a determiner, an optional modifier, and a head noun. They use their own system to cluster NEs with similar names, though they do not attempt any coreference on common nouns. They do not use pronouns, as they consider coreference resolution of pronouns to be the responsibility of the quote attribution system. In our previous results over LIT (O'Keefe et al., 2012), we identified pronouns automatically, and used a simple rule-based method to link them to either NEs or common nouns. Note that this corpus contains no gold-standard entities or coreference information, so for our experiments on LIT we do not report gold results, as they are not possible without further annotation.

Pride and Prejudice Corpus (P&P)

In P&P, He et al. (2013) found candidate speakers using the Stanford NE tagger, along with a manual preprocessing step where proper nominal mentions were clustered according to a small, manually produced set of aliases. He et al. consider a correct attribution to be from a quote to a character, rather than to a textually-grounded mention of a character. As such, their candidates only cover proper nouns, and do not consider pronominal or common nominal references. This means that they ignore any quotes that are attributed to unnamed characters, as they consider this to be out of scope.

4.1.2 Coreference Systems

In this chapter, we consider a supervised coreference system, an unsupervised system, and a baseline approach, which covers the range of approaches that exist for coreference resolution. The systems that we use are Stanford's CoreNLP package (Raghunathan et al., 2010), Reconcile (Stoyanov et al., 2010), and a naïve approach that clusters mentions based on simple string matching of head words (Hachey et al., 2013). In this work we attempt to run all of the systems with minimal deviation from their default settings. These systems are described in more detail in the following subsections.

Naïve Baseline

The naïve system is an amalgamation of the mention detection and proper name coreference found in Hachey et al. (2013) with simple rule-based pronoun coreference resolution, that links pronouns to the nearest gender-matching antecedent. The proper names are found using Stanford's NE tagger (Finkel et al., 2005), so that the naïve system uses will be consistent with those found using the Stanford and Reconcile coreference systems, which also use the Stanford NE tagger. The proper nouns are linked together via simple string matching rules. Pronouns are found using a lexicon and are joined to the closest gender-matching antecedent of the pronoun. The gender matches are determined through honorific titles, e.g. Mr. and Ms., and by using a name list based on the United States census data. The naïve baseline does not handle common noun mentions.

Stanford

Stanford's system (Lee et al., 2011) achieved the best result in the CoNLL 2011 shared task and remained competitive in CoNLL 2012 by using an unsupervised approach. It works by using a series of sieves that can each add mentions to coreference chains based on rules. The sieves are arranged in order of decreasing precision, so that mentions that are highly likely to be coreferential are clustered first.

Stanford's mention spans are by design longer than the other two systems, and include overlapping mentions. As our system is only able to handle non-overlapping mentions, we greedily kept the smaller mention of any overlapping pair, and retained the non-overlapping fragments from the longer mention as separate mentions. Some fragments and boundary tokens contained extraneous information, such as punctuation, which we removed. We also removed the part of any mention following a comma or WH word, so as to retain the head NP. This removes apposition and relative clauses, which makes the set of mentions more consistent for the quote attribution features. The default setting where all preceding mentions are potential antecedents was kept for the newswire corpora, but for LIT and P&P, a threshold of 100 sentences was used, due to the runtime.

Reconcile

Reconcile (Stoyanov et al., 2009) is a supervised coreference resolution system, whose authors aimed to create a modular, customisable system. It works by training a classifier to decide whether pairs of mentions are coreferential. During classification it can then produce a probability that each candidate mention is the antecedent. These classifications are then reconciled by taking the candidate with the highest probability, and setting it as the antecedent. By assigning an antecedent for each mention, the system will build clusters, which represent the end coreference chains. For features Reconcile uses token distance, string match, gender match, animacy, and others. Due to memory constraints, the longer of the LIT texts and the training set of P&P were processed in 500 paragraph chunks.

4.1.3 Extrinsic evaluation

In order to understand some of the problems that were occurring with the coreference systems, we examined some of the main cases of errors. The first problem we identify is that there are a large number of chains with a single mention whose token is POS tagged as a pronoun. Reconcile had the largest number of these with 13,938 (35% of the extracted pronouns) on LIT and 5,501 (33% of the pronouns) on P&P. This is consistent with the result in Kummerfeld and Klein (2013) which finds a large number of singleton mentions from Reconcile's output. This problem is particularly acute for quote attribution, as there are a large number of quotes that are directly attributed to pronouns.

Stanford does better on this problem, having only 1,238 singleton pronouns on LIT and 361 on P&P, of which only 154 and 43 are gendered. Stanford deterministically assigns pronouns to the closest compatible mention in the preceding three sentences and it seems that this is a better way of modelling pronouns in discourse. This is in line with the claim from Denis and Baldridge (2008) that the resolution of different mention types could be more successfully handled with a series of classifiers. However, of these 1,238 singleton pronouns, 549 are forms of you, which suggests that Stanford's discourse sieve needs to be extended to handle the complexities of literature.

Another major source of errors that we see when manually inspecting the data is conflation of chains corresponding to characters which share a family name, such as the "Miss Bennets" and their parents from P&P. To quantify this, we extract all the honorifics within a chain and report cases where a chain is assigned more than one honorific. For Stanford 1.7% of the mentions in LIT and 10.0% of the mentions in P&P are in chains with mixed honorifics, with the majority of the clashes coming from chains including honorifics for both genders. Reconcile makes a similar number of errors with 1.9% of mentions in LIT and 9.9% of mentions in P&P containing clashing honorifics.

4.2 Quote Extraction with Automatic Coreference Resolution

Our quote extraction experiments in Chapter 2 were all conducted using the gold-standard speaker candidates. While this information is not as central to quote extraction

	Strict				Partial			
	SMHC		PARC		SMHC		PARC	
	Node	Token	Node	Token	Node	Token	Node	Token
No mentions	75	78	65	68	86	89	84	84
Naïve	75	78	66	68	87	89	84	84
Stanford	75	78	62	66	86	89	80	82
Reconcile	75	78	62	67	86	89	81	82
Corpus	75	78	67	73	86	89	83	87

Table 4.2: Results of quote extraction experiments comparing no coreference or mentions, automatic coreference resolution, and the coreference that comes with the corpora.

as it is to quote attribution, it may still impact performance when automatic coreference chains are used. As such, we performed several quote extraction experiments with automatically detected and coreferred speaker candidates, and compared these results to experiments using gold-standard information. Note that neither of the two baseline approaches to quote extraction are affected by the presence of candidates, whether they are gold or automatic, so those experiments are not repeated.

Table 4.2 shows the results of these experiments. The first result that we highlight is that even without any mentions included, the quote extraction methods are still able to successfully detect quote spans. In particular, the accuracy on our primary corpus, the SMHC, is unaffected by the presence or absence of marked mentions. This is important as it shows that our quote extraction methods are not reliant on coreference resolution.

While the SMHC results are unaffected by the coreference information that is available, the PARC results are impacted somewhat. The impact largely falls on precision, rather than recall, and we speculate that the different training setup for PARC is the cause of the drop in performance. In particular, we speculate that the main cause of the drop in precision is that PARC is not fully labelled. This, coupled with our techniques to adjust for the incomplete training data, could cause the models to make too many predictions.

4.3 Quote Attribution with Automatic Coreference Resolution

Similarly to our quote extraction results, most of our quote attribution results in Chapter 3 used some gold-standard coreference information. While these results help us to get an idea of how effective our methods are, they are not completely realistic, as errors in the mention detection and coreference resolution can cause the quote attribution system to make mistakes. As such, in this section we will compare our results using the gold-standard information, with results using fully automatic mention detection and coreference resolution.

4.3.1 Speaker Alignment

In order to make this comparison we will need to link the gold-standard speaker annotations with the automatically detected mentions and coreference chains. Rather than conducting a full re-annotation process, which would be expensive and time-consuming, we instead define two separate alignment methods, which match coreference chains from the gold standard with automatically produced coreference chains. These alignment methods erase the gold-standard speaker annotation from each quote and replace it with one of the automatically generated coreference chains, so that the quote attribution methods can learn using automatically generated coreference chains only.

Both alignment methods work in roughly the same way. They find a mention in the gold-standard data that best represents the speaker of the quote, and they then find the coreference chain from the predicted data that contains that mention. Any speaker prediction to that chain would then be considered correct, and any quotes whose speakers could not be aligned would be considered incorrect, as no correct attribution is possible. In Section 4.3.2 we quantify the number of quotes whose speaker could not be aligned with a mention from an automatic chain.

Before discussing the methods in more detail, we would first like to note a weakness of our approach. For both of the alignment methods, if any coreference system outputs a single chain containing all mentions, it would get 100% accuracy. This occurs because the alignment methods simply overwrite each quote's speaker annotation with an automatic coreference chain, with no check to determine whether that coreference chain aligns to multiple gold-standard chains. As such, with a single predicted chain, any speaker prediction would be to the same chain and would be judged correct, as each quote's speaker would have been overwritten with the same chain. While this is not ideal, MUC F-score (Vilain et al., 1995) has a similar problem, so, as they do, we note that this evaluation cannot be considered independently of other metrics. In Section 4.3.3 we quantify this problem.

Source-based Alignment

Some of the corpora do not have gold-standard coreference chains, so our first alignment method aligns the gold-standard, textually-grounded source of each quote with a mention from the automatic coreference chains. Since speaker predictions are to whole coreference chains, any mention in the automatic coreference chain would be considered correct. Consider the following example:

> The Defence Secretary, Nick Warner, dismissed reports his department secretly investigated Mr Fitzgibbon, or held concerns that Ms Liu may have had links with China's military intelligence agency. [“We have found not a skerrick of evidence that there is any truth to these allegations,”]$_{quo}$ *he* said.

The textually-grounded source of this quote is the marked *he*, so the source-based alignment works by finding the automatic coreference chain that includes *he* as a mention. This automatic coreference chain would then be considered the speaker.

We use the annotated source for cases where a corpus includes an explicit source annotation, such as in SMHC and PARC. For the other corpora and quotes with no source, we align the automatic coreference chain with the mention from the gold-standard speaker's coreference chain that is nearest in word distance to the quote.

Canonical-based Alignment

Two of our data sets, namely SMHC and P&P, include full coreference information between the labelled gold-standard mentions, and have annotations of which gold-standard coreference chain represents the speaker. For these two corpora, we can use this information to assess whether the coreference systems are clustering the most canonical reference to each speaker with the source of the quote. We do this by setting the quote's speaker to be the automatic coreference chain that contains the most canonical mention of the speaker from the gold-standard annotations. The gold-standard canonical mention will normally be mentioned early in a document, and in the majority of cases will be an unambiguous reference to the entity. Consider the previous example, which is from early in a document and includes the first mention of *Nick Warner* in the document:

> The Defence Secretary, *Nick Warner*, dismissed reports his department secretly investigated Mr Fitzgibbon, or held concerns that Ms Liu may have had links with China's military intelligence agency. ["We have found not a skerrick of evidence that there is any truth to these allegations,"]$_{quo}$ he said.

In the example the most canonical form of the speaker is *Nick Warner*. So we would find the coreference chain containing the mention of *Nick Warner* and set it to be the speaker of the quote ("We have found..."). Ideally, the automatic coreference system would then cluster *Nick Warner* with the mention "he", which is the source of the quote. In this case the quote attribution system should predict the coreference chain containing "he", which would be clustered with *Nick Warner* by a perfect coreference system.

This alignment method allows us to determine how often the coreference system clusters the canonical form of a speaker with the form used to attribute a quote to the speaker. This is interesting as in many cases quotes have pronouns and common nouns as their source, and, as we previously noted, retrieving these sources as the speakers is not particularly informative.

4.3.2 Alignment Misses

Table 4.3 shows the percentage of quotes whose gold standard speaker had no corresponding mention in the automatic coreference chains. In general, the naïve approach had far more missing speakers, which is caused by its lack of common noun candidates. The naïve system performed particularly badly on the source-based alignment, as common nominal sources are frequent, while it performed better on the canonical-based alignment, where the mentions are largely proper nouns.

Between Stanford and Reconcile there is only a small difference in terms of the number of missed quotes. This indicates that both methods are correctly generating most of the

	Source				Canonical	
	SMHC	PARC	LIT	P&P	SMHC	P&P
Naïve	14.2	23.0	11.6	1.0	3.1	3.3
Stanford	1.3	0.2	0.6	0.5	0.1	4.3
Reconcile	1.2	0.1	1.5	0.5	1.7	0.8

Table 4.3: Percentage of quotes in the training set that have gold speakers that could not be aligned to automatic speakers.

	Source		Canonical	
	SMHC	P&P	SMHC	P&P
Naïve	2.7	68.3	0.3	33.5
Stanford	0.9	22.1	0.9	4.3
Reconcile	2.0	36.2	1.6	24.9

Table 4.4: Percentage of quotes whose automatic speaker chain aligned with mentions from multiple gold-standard chains. Results for PARC and LIT are not available as they do not have full coreference information.

required mentions, with both methods performing particularly well with the canonical-based alignment. It is also worth noting that these quotes would not necessarily be impossible to classify correctly, as non-source or non-canonical mentions of a quotes' speaker may be present in the automatic coreference chains.

4.3.3 Overlapping Chains

Table 4.4 shows the percentage of quotes whose automatic coreference chain aligned to mentions from more than one gold-standard chain. In these cases, the coreference systems have over-clustered the chains, such that even if the quote attribution system predicts the correct chain, the speaker would be ambiguous due to the over-clustering. While this is a problem for evaluation, we note that for the SMHC it occurs infrequently. On the SMHC, the worst-performing system, Naïve, over-clusters 2.7% or fewer quotes. Reconcile over-clusters the speakers of 2% or fewer quotes, while Stanford does even better with 0.9% of

	SMHC			PARC		
	Rule	No seq.	Greedy	Rule	No seq.	Greedy
Naïve	72	81	81	57	69	69
Stanford	68	89	89	69	85	84
Reconcile	71	87	87	67	85	84
Corpus	85	92	92	78	97	97

Table 4.5: Quote attribution results on the news corpora using the source-based alignment method. The corpus results use the candidates that come with the corpora.

quotes over-clustered. As the scale of this problem is small for the SMHC, we ignore it in evaluation.

By contrast the results on P&P are very poor, with frequent over-clustering. In P&P, there are many cases where multiple family members are referred to by their family name and an honorific. While the honorifics make it clear that the characters are distinct, the coreference systems are ignoring the honorifics and clustering the characters together. He et al. (2013) chose to manually create a set of aliases, which fixed this problem. The results we present here validate that choice, as the coreference systems are not able to work effectively with P&P. Despite the poor performance by the coreference systems, we present results on P&P, with the caveat that further disambiguation would be required to identify the correct speaker.

4.3.4 Attribution Results

In Chapter 3 we evaluated several methods of attributing quotes to speakers. While the different methods achieved quite varied results across some of the corpora, they were very close on our primary corpus, the SMHC. As such, for the evaluations in this chapter we will present results using only three of the approaches. The first approach is the rule-based system, as it tells us how far we can get without a learning stage. The second and third approaches are the binary class model with no sequence features and the binary class model with greedy decoding of the sequence features.

	LIT			P&P		
	Rule	No seq.	Greedy	Rule	No seq.	Greedy
Naïve	43	48	48	65	67	67
Stanford	36	47	48	40	50	52
Reconcile	32	45	48	40	67	68
Corpus	55	64	64	55	79	84

Table 4.6: Quote attribution results on the literature corpora using the source-based alignment method. The corpus results use the candidates that come with the corpora.

The source-based alignment results for the news corpora are shown in Table 4.5. In almost all cases, the coreference systems were able to help the quote attribution systems when compared to the naïve baseline. In particular, the learned methods are able to learn to avoid common nominal references, which few quotes are attributed to. By comparison, the rule-based system frequently chooses to attribute quotes to common nouns, which means that it performs comparatively poorly.

Both Stanford and Reconcile achieved clearly better results than the naïve baseline. On the SMHC, the Stanford system improved over the naïve baseline by 8%, while Reconcile improved over it by 6%. On PARC, the results are even more impressive, with both Stanford and Reconcile improving over the naïve baseline by 15-16%. This indicates that good coreference information is important for quote attribution on news corpora.

While both Stanford and Reconcile improved over the naïve baseline, there is little difference between their performance. On the SMHC the Stanford system produced the best results, with 89% for both learned methods, while Reconcile achieved 87% for both. On PARC, the two systems performed almost equivalently.

Table 4.6 shows the results with the source-based alignment on the two literature corpora. Unlike the news corpora, where the coreference information had a clearly positive impact, the results here are more mixed. For LIT the best result is obtained by using the greedy quote attribution system. However, this result is equivalent for all of the automatic coreference systems, meaning that the more advanced methods used by Stanford and Reconcile had no impact. Furthermore, the result is 16% lower than the mentions and

	SMHC			P&P		
	Rule	No seq.	Greedy	Rule	No seq.	Greedy
Naïve	51	67	67	66	73	70
Stanford	40	56	57	29	34	39
Reconcile	40	59	58	35	47	55
Corpus	85	92	92	55	79	84

Table 4.7: Quote attribution results using the canonical-based alignment method. The canonical alignment is only possible on corpora that have full coreference, so we only show results for SMHC and P&P.

coreference that are included with the corpus. As the LIT mentions and coreference use no gold-standard information, these results validate Elson and McKeown's (2010) claim that coreference systems do not perform well over literature.

On P&P, the naïve system and Reconcile achieved largely similar results, with the single best result produced by using Reconcile with greedy decoding. Both of these systems clearly outperformed the Stanford system, however this is potentially misleading. In Table 4.4, we showed that both Reconcile and the naïve baseline suffered from an over-clustering problem. This metric is unable to penalise systems for over-clustering, so the results from Reconcile and the naïve system may be quite optimistic. The Stanford system suffers least from the over-clustering problem, but even its results are somewhat unreliable, as 22.1% of quotes are attributed to chains that align to multiple gold-standard chains.

It is worth noting that using the mentions and coreference that were provided with P&P results in 32% better performance than the best result using the Stanford system. The key difference between the two is that He et al. manually built a small set of aliases for each character. That this causes a difference as large as 32% is indicative of how important good coreference information is for this task.

The results for the canonical-based alignment methods are shown in Table 4.7. On the SMHC, the canonical results are worse than the source-based results. This may be problematic as it could mean that a canonical form of an entity cannot be found in the automatic coreference chain that is predicted to be the speaker. In other words, although the quote attribution system has linked the quote to a coreference chain, there might not be a canonical form of the speaker in that chain. These low results are particularly surprising as the coreference systems are trained on news corpora.

While these results are surprising, they may be misleading for two reasons. The first is that although the results are quite low, they do not necessarily indicate that it is impossible to identify *any* meaningful form of the speaker in the automatic coreference chain, only that the *most* canonical form does not appear. The second is that errors in the coreference step will negatively impact the training process, as the source of each quote may not be in the correct coreference chain, and so might not be used as a positive instance during training. The training process would then presumably create less consistent models that tend to make more mistakes.

On P&P, Stanford and Reconcile produced results that are worse than both the Naïve results and the results using the corpus-provided mentions and coreference. In particular, there is a large gap between the Stanford results and the results using either Reconcile or the Naïve system. We speculate that, similarly to the source-based results, this is caused by the Naïve system and Reconcile over-clustering mentions, particularly those with the same family name.

Overall the results of these experiments show that coreference resolution removes some of the inconsistencies between the corpora, which allows for a more realistic comparison of results. The source-based alignment results are only slightly worse than using the mix of gold standard and automatic coreference that is available in the corpora, with some results that are better using fully automatic coreference resolution. The challenge with the source-based alignment is that it does not necessarily yield clusters that contain an informative reference to the real-world entity. The canonical-based alignment results, which address this problem, show that canonical mentions are not always clustered with all of their later references, which is problematic for quote attribution. Fully quantifying this problem would require fully-annotated coreference data, which is currently unavailable. Nonetheless, our results show that automatic coreference resolution is somewhat successful in news data.

4.4 Full Pipeline

In our previous evaluations of quote attribution we took the actual spans of the quotes as given. For the two literature corpora, which only contain direct quotes, this is a reasonable assumption, as extracting direct quotes is relatively straightforward. However, for the two news corpora the mixed and indirect quotes are more challenging to extract, as shown in Chapter 2. In this section we will evaluate our quote attribution methods in conjunction with automatic quote extraction, so that we can gauge the impact of automatic quote extraction. We also note that as the quotes in the literature corpora are relatively easy to extract, the results on those corpora would be the same as in the previous section, so we will only present results on the news corpora.

As we are attributing quotes to automatic coreference chains we need to use the alignment methods from the previous section, so that each quote has a correct answer. This presents an issue for predicted quotes that do not exactly match a quote from the gold standard, as a single predicted quote might match multiple gold quotes with different speakers. In order to keep the evaluation simple and easy to interpret, we count any mismatched quote as incorrect, and focus only on the predicted quotes that exactly match a quote from the gold standard. Like in previous sections we will use our most effective quote extraction method, the token-based approach.

In this evaluation, we cannot simply use accuracy as our evaluation metric, as the set of quotes that we are attributing will be of varying size, depending on the success of the quote extraction process. Instead we use precision, recall, and F-score, with correctness defined as the number of quotes whose span we have exactly matched and whose speaker we have correctly predicted. As in earlier sections, we consider the speaker to be the coreference chain, rather than an individual mention of an entity. If we consider $correct$ to be the number of predicted quotes that exactly match a span and speaker from the gold standard, $predictions$ to be the number of predictions, and $gold$ to be the number of gold-standard quotes, then precision, recall, and F-score are:

$$P = \frac{correct}{predictions}$$

	SMHC			PARC		
	P	R	F	P	R	F
Naïve	66	63	65	41	62	49
Stanford	72	69	70	50	77	60
Reconcile	71	68	69	49	76	59
Corpus	68	64	66	53	79	63

Table 4.8: Results for the full pipeline of mention detection, coreference resolution, quote extraction, and quote attribution, with the coreference chains aligned using the source-based method. The quote extraction used is the token-based approach and the quote attribution is the greedy approach.

$$R = \frac{correct}{gold}$$

$$F = \frac{2PR}{P + R}$$

The results using the source-based alignment are shown in Table 4.8. Precision and recall on the SMHC are fairly balanced, lying in a narrow range between 63% and 72% across all methods. This indicates that for the SMHC, the combination of methods applied has made a reasonable trade-off between coverage and correctness. For PARC, the precision and recall are not well balanced, however, with precision lying in the range of 41% to 53% and recall lying in the range of 62% to 79%. Again this difference is likely caused by the incomplete data that is available for PARC.

The SMHC results show that using automatic coreference resolution makes some improvements over both the gold-standard annotations and the naïve system, which both lack common noun candidates. Our systems using coreference from Stanford and Reconcile achieved very similar results, with F-scores of 68% and 67% respectively. These results are significantly lower than the best result with gold standard quote spans, which is 89%. However, they compare well to previous work (Schneider et al., 2010), which achieved an F-score of 54% for a similar task. Our results on PARC are somewhat lower, with F-scores of 60% and 59% for Stanford and Reconcile respectively.

The final results that we present in this section are from using the full pipeline with the canonical-based alignment method. As the canonical-based method requires full corefer-

	SMHC		
	P	R	F
Naïve	54	52	53
Stanford	47	45	46
Reconcile	48	45	46
Corpus	68	64	66

Table 4.9: SMHC results for the full pipeline of mention detection, coreference resolution, quote extraction, and quote attribution, with the coreference chains aligned using the canonical-based method. The quote extraction used is the token-based approach and the quote attribution is the greedy approach.

ence information, we cannot use PARC for these experiments. The results for the SMHC are shown in Table 4.9. Similarly to the findings in the previous section, using the canonical-based alignment method tends to have a negative impact on the results, with F-scores between 46% and 53%. Interestingly, the naïve baseline actually produces better results than the two coreference systems, which is likely caused by the two coreference systems avoiding clustering common noun mentions with proper noun mentions. This problem does not affect the naïve baseline, which does not use common noun references.

Overall the best result using the canonical-based alignment is 17% lower than the best result using the source-based alignment method. This confirms our previous finding that the coreference systems are often failing to cluster the most canonical reference to an entity with other mentions of the entity in question. As we noted previously, these results do not tell us whether there is a useful, i.e. not pronominal or common nominal, mention in the attributed speaker's coreference chain. While this is far from perfect, getting results that more robustly tell us how often we can attribute a quote to a useful mention of its speaker would require a new round of annotation for each coreference system, which would be time consuming and costly.

In lieu of performing this reannnotation, we present results in the next section that use the full pipeline that we have introduced thus far, with an additional step that uses an NEL system to link coreference chains to a KB. While these results are only possible on the SMHC, they will tell us what proportion of quotes can be attributed unambiguously to the correct KB entry.

4.5 Full Pipeline with Named Entity Linking

Our previous experiments have given us insight into the impact of coreference resolution on both quote extraction and attribution, and allowed us to use quote attribution as an extrinsic evaluation of coreference resolution, they still have not given us a definitive answer about how many errors we should expect as input to our opinion mining system. In this section, we use a NEL system with the coreference resolution systems to answer that question. Though previous experiments were possible over multiple corpora, we can only perform these experiments over the SMHC, as it is the only corpus that has speakers that have been fully disambiguated back to a KB. In the next section we will introduce the task of NEL and the system we use, followed by a description of the experimental setup and results.

4.5.1 Named Entity Linking (NEL)

NEL is the task of linking mentions of entities, such as people, organisations, places, etc., to nodes representing those entities in a KB, or to NIL if there is no such node. NEL systems need to handle the issues of *ambiguity*, where two or more entities share a name, and *synonymy*, where there are multiple ways to refer to a single entity. In general, NEL systems address this by breaking the task down into several subproblems (Hachey et al., 2013). The first is *extraction*, where the named entities are detected and coreference resolution is performed. This is followed by a *search* phase, where a set of candidate KB entries are found for each mention (or coreference chain). The final step is *disambiguation* where the system selects one KB node from the set of candidates, or if there are no appropriate candidates then the mention is marked as NIL. Most NEL systems work by using only proper nouns, as it is difficult to make use of the disambiguating information that is provided by pronominal or common nominal references to entities.

The NEL system that we use is the unsupervised model from Radford et al. (2013) with a Wikipedia dump[1] from April 2012 as the KB. As its extraction process the system uses the C&C Tools (Curran, 2004) NE tagger and the naïve coreference system described earlier, albeit without the pronouns. For the search phase the system uses Solr[2] with a set of aliases for each entity from Wikipedia, the Crosswikis data (Spitkovsky and Chang, 2012). The final disambiguation step is performed in an unsupervised way by averaging several scores for each candidate. The scores are the prior probability of linking to the candidate, the probability that a given alias links to the candidate, the overlap of categories between the candidate and other links in the document, the normalised overlap of the inlinks to the otherwise top-ranked entities across the whole document, and the count of supporting entities near a mention (e.g. finding Ontario near Melbourne, would suggest that we should link to Melbourne, Ontario rather than Melbourne, Victoria).

We use the above system with all default settings with one exception. The NE tagger that is built into the system was trained on some of the same articles that we have as our SMHC test data, so it should perform unreasonably well on those documents. The system also does not detect any common nouns or pronouns, and so has no coreference information between those items. To address this we perform experiments using all of the coreference systems that were described earlier in this chapter. This is done by first using the coreference systems to generate and corefer all mentions. The NEs are then passed to the NEL system, which disambiguates these mentions back to KB entries. For our quote extraction and attribution experiments, we add the common nouns and pronouns back in, so that the full set of mentions is available to these systems.

4.5.2 Experimental Setup

In Section 4.4 there were two key findings. The source-based alignment results showed that many quotes are attributed to their speaker via common nominal references, and including these references as candidates can improve performance. Conversely, the canonical-based alignment results showed that the common nominal references are rarely clustered together with the most canonical form of the speaker, which means that the coreference

[1] http://dumps.wikimedia.org

[2] http://lucene.apache.org/solr

chain that is returned as the speaker might not be very informative. The first finding is an argument for the inclusion of common nouns, while the second is an argument against their inclusion.

In this set of experiments we run the NEL system with only the proper nominal mentions, as the NEL system is not designed to handle pronominal or common nominal references. We then merge the pronominal and common nominal mentions back in, such that there is a full set of mentions available to the quote attribution system. This means that the quote attribution system can assign a quote to a speaker that has been disambiguated back to a KB by the NEL system, or to a speaker that is just a standard coreference chain with no such disambiguation. We consider this to be necessary, as excluding the common and proper nouns would create too many false positives, i.e. quotes that are mistakenly attributed to proper nouns because the correct pronominal or common nominal mention is not available. The drawback of this approach is that we will mistakenly assign some quotes to common nominal or pronominal candidates.

This issue has an impact on how we evaluate our approach, as we have a mix of quotes that have been attributed to disambiguated candidates and those that have not been disambiguated. In our opinion mining work, we are most interested in the quotes that have been attributed to known entities, so our evaluation metrics will focus on those quotes. For this set of metrics we define a correct result to be a quote whose span and disambiguated entity matches a quote from the gold standard ($correct$). The set of gold standard quotes will be only those quotes that are disambiguated back to a KB ($gold$), and the set of predicted quotes will be only those quotes whose predicted speaker has been disambiguated to a KB ($predictions$). We can then define precision, recall, and F-score as follows:

$$P = \frac{|correct|}{|predictions|}$$

$$R = \frac{|correct|}{|gold|}$$

$$F = \frac{2PR}{P + R}$$

We also consider a variant of the partial score that was used in Chapter 2. This scoring works in the same way, with the key difference that for the $overlap(x, y)$ function, we only consider spans to be overlapping if their disambiguated speakers match.

	Strict			Partial		
	P	R	F	P	R	F
Naïve	31	51	38	34	58	43
Stanford	31	42	35	33	47	39
Reconcile	32	41	36	36	45	40
Gold coreference	37	66	47	40	74	52
Gold coref. + links	66	71	68	72	80	76

Table 4.10: Results of running the full pipeline with NEL.

This set of evaluation metrics is able to capture how well our system does when we can link a quote to a KB entry. The problem with this is that for the remaining quotes we get no information about correctness at all. While it would be ideal to have a more complete evaluation, in order to achieve such an evaluation we need to either fall back to our alignment methods, as we did in Section 4.4, or to reannotate the entire corpus for each coreference system. Reannotating the entire corpus is simply too expensive and not robust to new methods, while the alignment methods introduce the same issues as noted in Section 4.4. We further note that most of the opinions that we will be interested in are from entities that appear in the KB, so for these experiments we limit ourselves to the evaluation as described.

This limitation, where correctness is judged entirely with reference to the entries in the KB, has several implications. The first is that achieving wider potential coverage is possible by simply adding new nodes in the KB. Secondly, although there will be quotes that are attributed to speakers that are linked to NIL, the attributions may still be useful, depending on the application. For instance, it is easy to imagine an opinion mining system that compares opinions from known entities to those of unknown entities.

4.5.3 Results

Table 4.10, shows that with gold-standard mention detection, coreference resolution, and entity disambiguation, the quote extraction and attribution systems achieve reasonable performance, with a strict F-score of 68%. This is not far below the F-score of 75% for quote extraction alone, which indicates that quote attribution in particular is performing well. The partial F-score, which allows for partial matches of quote spans, is also very strong at 76%.

However, when we remove the gold-standard entity disambiguation and rely on the NEL system, the strict F-score drops to 47%, which is mostly due to a large drop in precision, from 66% to 37%. The main difference between the output of these two configurations of the system is that when gold-standard disambiguation is used the system predicts that there are 1,180 quotes attributed to linked entities. By contrast when automatic NEL is used there are 1,934 quotes attributed to linked entities. Given that the NEL system is recall-focused (Radford et al., 2013), this is unsurprising.

When we move from using gold-standard mentions and coreference information the results drop even further, although in this case it is more from a drop in recall. Stanford and Reconcile have approximately equal strict F-scores of 35% and 36% respectively, with precision and recall also fairly even between them. The naïve system performed comparatively well, with an equivalent precision score but with recall that is almost 10% higher. This is potentially due to the NEL system being optimised for the output of the naïve system.

4.6 Summary

In this chapter we have performed a complete, realistic, and end-to-end evaluation of quote extraction and attribution. A core part of this evaluation involved examining the impact of three different coreference resolution systems. Though there is no straightforward evaluation metric, we ran the three coreference systems over the four corpora, and aligned the gold-standard speaker annotations with the chains produced by the coreference systems. When combined with the output of the quote extraction system, this provides a realistic setting for evaluating those methods, as no gold-standard mentions, coreference information, or quote spans were required.

The results show that quote extraction is unaffected by the choice of coreference system, and even performs equivalently with no mentions or coreference information at all.

Evaluating quote attribution with automatic coreference resolution is more difficult. Our results show that we can attribute a quote to the chain representing its *source* with up to 89% accuracy on the SMHC. Despite this encouraging finding, we also show that coreference resolution systems often put the source of a quote in a different chain to the most canonical representation of its speaker, which may hinder the utility of these systems.

While the coreference systems performed reasonably well on the two news corpora, there were issues with their output on the two literature corpora. The single biggest problem was that the coreference systems were too aggressive in merging together chains, and so tended to produce relatively few chains, with multiple entities in each chain. Literature contains more entities with familial relations, and these entities will share family names. Surprisingly, both coreference systems frequently merged chains representing distinct family members, even when conflicting honorifics were present.

With these results in place, we moved on to examine the effect of running the full pipeline of mention detection, coreference resolution, quote extraction, and quote attribution. Evaluating the full set of systems has the same problem as evaluating quote attribution with automatic coreference resolution, however the resolution is equivalent. Our results show that we can attribute quotes to the chain containing their sources with a strict F-score of 70%, though this score drops to 53% when we consider whether the chain includes the most canonical mention of the speaker.

In our final set of experiments, we added another layer to the pipeline: Named Entity Linking. Adding NEL allows us to link quotes to a KB, such as Wikipedia. However, we found that this extra step produced fairly mediocre results. On the SMHC, the best strict F-score that we achieved was 38%, which was largely due to poor precision (31%).

Chapters 2 and 3 evaluated a range of novel methods for quote extraction and attribution, on a range of corpora, and with explicit comparisons to previous work. In this chapter, we have expanded on this further, by evaluating the full pipeline of systems. Crucially, these evaluations tell us how good the input to the opinion mining work should be. In the next chapter we will conduct an overview of previous work in opinion mining.

5 · Opinion Mining

In many areas of our lives it is useful to know the opinions of others. In online shopping for instance, a shopper may be less inclined to purchase a product if the reviews of the product are uniformly unfavourable. Similarly, while reading the news it may be informative to know the past and present opinions of salient people, so as to provide further context to the story. Systems that find these opinions and classify them are called opinion mining or sentiment analysis systems.

A recent survey (Pang and Lee, 2008) showed that there have been hundreds of academic publications on the topic, as well as significant media and commercial attention. With this level of interest, many approaches, corpora, and task definitions have been proposed, which has made the field quite diverse. As such, there are a number of subtasks within opinion mining, with some being more related to our work than others.

In this chapter, we will provide an overview of opinion mining as a whole, and discuss how some of the subfields relate to our work. We will begin in Section 5.1 by covering work on building lexicons of sentiment-bearing words. Next, in Section 5.2, we will discuss document-level sentiment analysis. In Section 5.3, we will look at the related work in subjectivity detection, where the aim is to distinguish opinion-bearing text from objective or factual text. In Section 5.4 we will look at work that breaks opinions about an entity down into opinions about individual aspects of that entity. Next, we move on to discuss opinion mining in news and politics in Section 5.5, with a particular focus on political stance classification. We will then discuss some work on Opinion Question Answering (OPQA), which is concerned with making traditional Question Answering (QA) systems opinion-aware.

5.1 Sentiment Lexicons

Much of the early work on opinions within the NLP community was focused on building lexicons of words marked with their *semantic orientation* – later known as *polarity* – (Hatzivassiloglou and McKeown, 1997). The idea of this task is that some words carry broadly *positive* connotations, such as "good", "happy", and so on, while other words carry broadly *negative* connotations, such as "bad", "anger", and so on.

In their work, Hatzivassiloglou and McKeown (1997), built a lexicon of adjectives that were scored with their semantic orientation. In other words, the adjectives had been rated in how *positive* or *negative* the sentiment they evoke is. They accomplished this by using the fact that adjectives in coordinating conjunctions tend to have the same polarity. For example, the authors noted that it would be valid to describe elections as "fair and legitimate" or "corrupt and brutal", but not "corrupt and legitimate". They used a log-linear classifier to predict whether adjectives that are in a coordinating conjunction have the same semantic orientation. They then clustered the adjectives based on this result so that they had two groups of adjectives, with one marked *positive* and the other *negative*.

While Hatzivassiloglou and McKeown's work was able to detect the semantic orientation of adjectives, it does not work for other parts of speech. Work by Esuli and Sebastiani (2005) addressed this by finding the semantic orientation of terms by looking at their gloss, which is the term's definition as found in a dictionary, in addition to links with other words found in a thesaurus. They used WordNet (Fellbaum, 1998) as their dictionary, as it contains both glosses and relations between words, which Kamps et al. (2004) had already shown were useful for determining the semantic orientation of adjectives. The authors used the synonym, antonym, indirect antonym, hypernym, and hyponym relations to expand their initial seed sets of positive and negative words. They then trained bag-of-word classifiers on the glosses of these sets and used them to classify the remaining words. Their methods achieved accuracies of between 83.09% and 88.05% on existing, manually-built lexicons.

Esuli and Sebastiani later extended their work to create SentiWordNet (Esuli and Sebastiani, 2006b). In creating SentiWordNet they assigned a *positive* score, a *negative* score, and an *objective* score to each synset in WordNet, where the three scores sum to one. They followed the same idea as in their previous work, but used a committee of eight classifiers to make the classifications. These eight decisions were then reconciled by having each classifier assign exactly one of the labels, and dividing the counts of each label by eight to get the final score. Later work updated SentiWordNet in various ways (Esuli and Sebastiani, 2006a, 2007b,a), with the most recent work (Baccianella et al., 2010) updating SentiWordNet to use WordNet 3.0, along with some improvements to the method.

Other researchers have looked at the problem of detecting terms with domain-specific polarity (Andreevskaia and Bergler, 2006; Kanayama and Nasukawa, 2006; Choi and Cardie, 2009), while others have investigated attaching polarity to unseen words (Moilanen and Pulman, 2008) through the assumption that certain morphemes carry sentiment.

As we will see in the next few sections, sentiment lexicons have primarily been used as input to other opinion-related tasks. In fact, most methods make use of a sentiment lexicon in some form. Surprisingly though, they are not as relevant to news text, as opinion-bearing words are relatively infrequent.

5.2 Document-level Classification

For document-level sentiment classification, Turney's (2002) is a key study. Turney aimed to classify movie reviews as either *positive* or *negative* about the item under review, based on the adjectives that appeared within the document. The classification worked by calculating the semantic orientation of adjectives using Pointwise Mutual Information (PMI) – which is a measure of similarity – between each adjective and the words "excellent" and "poor". The adjective would then be scored by subtracting the PMI between itself and "poor" from the PMI between itself and "excellent". The whole review was then classified by summing up the scores of the adjectives within it. When evaluated on a corpus of movie reviews Turney's approach yielded an accuracy of 65.83%.

While Turney's approach has the advantage of being unsupervised, and thus easily generalisable to new domains, it has some significant drawbacks. Chief among them is that Turney used search engine hits to calculate the PMI, as he required estimates of how often two words occurred together and how often the two words appeared at all. While it proved effective for his initial work, the search engine step made Turney's original approach impractical, although later refinements (Turney and Littman, 2003) alleviated this problem.

Turney's work was closely followed by the work of Pang et al. (2002), who similarly aimed to classify movie reviews as being either positive or negative about the movie they review. They treated the task as a special case of topic-based text categorisation where the topics are replaced with the labels *positive* and *negative*. As such, they evaluated several classifiers that were successful in text categorisation, namely Naïve Bayes (NB), Maximum Entropy (ME), and Support Vector Machine (SVM). For features they evaluated unigram and bigram bag of words, with the feature values being either binary-valued or word frequency. These approaches were evaluated on a corpus of movie reviews that included a score of the movie from the original reviewers, which removed the need for manual annotation. Pang et al. found that their classifiers performed best with a unigram bag-of-words model, and that using binary features vastly outperformed word frequency. In replicating their results, we found that deleting stopwords removed this difference. Their best accuracy was 82.9%, which was achieved using a SVM. In later work, (Pang and Lee, 2004) the authors used sentence-level subjectivity detection to remove objective sentences from consideration. This caused a drop in accuracy in their initial experiments, but by adding a weighting so that adjacent sentences tended to receive the same classification they were able to achieve slight improvements in accuracy.

While both Turney's work and Pang et al.'s work used movie reviews as data, they both noted that the methods were equally applicable to other types of reviews. Blitzer et al. (2007) established this experimentally by creating new data covering four types of products. With this data they also investigated the domain adaptability of the text categorisation methods, by testing how well a model trained on one product would work on a different product. They found that in most cases accuracy dropped by 10% or more, although they proposed the use of the Structural Correspondence Learning algorithm (Blitzer et al., 2006) to somewhat ameliorate this loss. Work presented by Abbasi et al.

(2008) covered multiple languages, and showed that feature selection can yield state-of-the-art results.

Work by Taboada et al. (2011) introduces Semantic Orientation CALculator (SO-CAL), which takes a lexicon-based approach to review classification. Like Blitzer et al., the authors noted that the domain adaptability of learned classifiers was relatively poor. They found that this was because the classifiers were learning some very domain-specific features, which were misleading when used in a different domain. By using lexicons, they argue that their method is able to avoid these issues, as the method is not trained with respect to a single domain. The SO-CAL system thus relies on manually-constructed lexicons of adjectives, nouns, and verbs, as well as an automatically-built list of adverbs. These words are combined with valence shifters and some text-level features to produce a single score of semantic orientation. Their evaluation shows that their method is competitive with text classification approaches, and is comparatively domain independent.

Other work focused on classifying the sentiment in documents on a more fine-grained scale than simply positive and negative. Pang and Lee (2005) followed on from their earlier work by classifying movies on both three and four-star scales. In the three-star scale, one star represents bad, two represents "middling", and three represents good. The authors noted that previous work (Koppel and Schler, 2006)[1] treated the middle label as neutral, while they treat it as a combination of neutral and mixed sentiment. The four-star scale had no explicit neutral rating, with one star representing bad and four stars representing great. Their paper showed that human agreement for opinion related labelling schemes can be low, even for schemes that we may assume humans can consistently interpret, such as a five-star rating scheme. It also showed that as we increase the number of labels (stars in this instance), the difference between the baseline result and learned methods decreases.

Document-level opinion mining is an interesting task, but in some ways it has limited utility. The basic set up of the task is valid for any set of documents where the content is assumed to be opinionated, and where a limited set of labels, typically two or three, can be used to categorise the opinions within the document. While we have discussed the

[1] Pang and Lee cite an earlier version of this paper.

applicability of document-level sentiment classification to movie and product reviews, there are many areas where a single label per document is far too simplistic. In particular, in news text it is likely that there will be multiple opinionated sources quoted, which removes the utility of a single document-wide label.

This problem was realised early on, with several authors proposing tasks and approaches that worked over smaller textual units. Before we discuss these publications however, we will discuss the related field of subjectivity detection.

5.3 Subjectivity Detection

Subjectivity detection is the task of determining whether a given unit of text is *subjective* or *objective*. Broadly speaking, subjective in this context means that the text contains an opinion or evaluation of something, while objective means that it is presenting something factual.

Subjectivity detection is clearly related to opinion mining, as it aims to detect opinionated content. The key difference is that in opinion mining, the goal is to categorise opinions according to some labelling scheme, such as by labelling them as positive or negative about some given product. By contrast, in subjectivity detection the goal is to distinguish between text that is objective, or factual in nature, and text that is subjective, or opinionated. It may seem as though subjectivity detection is a necessary preprocessing step for opinion mining. However, in many cases opinion mining is conducted over text that is known or assumed to be opinionated, making subjectivity detection unnecessary. There are also many applications, such as QA, where it can be important to know that text is subjective, but where it does not matter what the polarity of the text is.

Early work in the field was concerned with tracking point of view in narratives (Wiebe, 1990, 1994; Wiebe and Bruce, 1995). The goal of these studies was to find the entity whose point of view was being represented in subjective sentences. This is in some ways similar to our work in attributing quotes, except that it operates on a whole sentence level, and does not require a speech event to take place. While these studies needed to identify subjective sentences, it was not the focus of their work.

Bruce and Wiebe (1999) were the first to demonstrate that it is possible to label subjective sentences from news articles with reasonable inter-annotator agreement. In Wiebe et al. (1999), the authors trained a sentence-level classifier using simple syntactic features, which achieved 72.17% accuracy, compared to averaged annotator agreement of 89.5%. Subsequent work by Hatzivassiloglou and Wiebe (2000) examined how well the presence of certain adjectives could predict whether a sentence was subjective or objective, notably including the adjectives built in Hatzivassiloglou and McKeown (1997).

Work by Wiebe (2000) further examined this link between subjectivity and adjectives. Wiebe used the distributional similarity corpus developed by Lin (1998), along with seed sets of adjectives from the training set of their corpus to bootstrap a larger set of adjectives that are likely to be subjective. Using just these adjectives to identify subjective sentences achieved an average precision of 61.2%, which is a substantial improvement over the 55.8% they achieved using just the seed set of adjectives. They then continued to evaluate the effect of using the intersection of each of these sets with the polar adjectives from Hatzivassiloglou and McKeown (1997), as well as the gradable adjectives from Hatzivassiloglou and Wiebe (2000). These techniques allowed them to get even greater average precision, ranging from 60.4% to 79.6%, with reasonable coverage. These methods all used adjectives to identify subjective expressions, however this limits their potential coverage to subjective sentences that actually contain adjectives.

Work by Riloff and Wiebe (2003) aimed to address this by identifying subjective sentences through the use of extraction patterns (Riloff, 1996). Extraction patterns are constructs that can be used to extract phrases of interest. They include some fixed words and some wildcards that may be filled by words or phrases, which makes them able to extract things that n-grams cannot. This allows them to capture instances of non-compositional meaning, such as the pattern[2] "<x> drives <y> up the wall", where <x> and <y> can be filled by any noun phrases, rather than just single words. For instance this pattern could capture "George drives me up the wall" as well as "the nosy old man drives his neighbours up the wall". The extraction patterns are learned via a bootstrapping procedure that requires a small amount of labelled data. Using the extraction patterns improved the

[2] Example taken from Riloff and Wiebe (2003)

coverage of their high precision subjectivity classifier from 32.9% to 40.1%, with a minimal impact on precision.

In later research, Riloff and Wiebe (2003) used a boostrapping approach with extraction patterns to identify subjective nouns. They did this by looking for nouns matching empty slots in patterns, which they then used to find more patterns that included these nouns, which allowed them to find more nouns, and so on. They used the final set of nouns, along with features from their earlier work to build a NB classifier that achieved 81% precision and 77% recall. Later work by Wiebe and Riloff (2005) extended this work to include objective extraction patterns. They evaluated their work on much more data and found that they could substantially increase recall over their previous approaches, with a minimal loss of precision.

The work in subjectivity detection that we have discussed thus far has focused on classifying the subjectivity of sentences. Wiebe et al. (2001) went further to try to identify whether news articles were editorials or not. They based their method on detecting collocations, with the unusual feature that they allowed certain words in a collocation to be replaced by infrequently occurring words, as they found that opinionated text tends to use more low-frequency words than objective text. They detected collocations by finding all instances of 1 to 4-grams that met certain precision requirements in detecting subjective text in their training data. They then employed their idea about infrequent words by replacing all words that appeared once in their training data with a special token. This enabled them to rerun their collocation detection method to find a wider set of collocations. Finally they use this, plus other features in a linear regression model to classify news articles as editorials or not editorials. Their methods achieved an accuracy of 91.2%, compared to a baseline of 89.2%.

5.4 Aspect-oriented Opinion Mining

While document-level sentiment analysis classifies whole documents as positive or negative, aspect-oriented opinion mining looks at extracting and classifying smaller units of text that express sentiment towards some entity or aspect of an entity. Early research into this problem looked at making classifications on a per-sentence level, mainly to support other tasks, such as information extraction. Yu and Hatzivassiloglou (2003) predicted the subjectivity of sentences using several NB classifiers. They then found the semantic orientation of these subjective sentences using the average semantic orientation of the terms within them.

Dave et al. (2003) were the first to explicitly identify aspects of products, although it was not the main focus of their paper. Their paper focused on evaluating the effects of features and preprocessing changes to the document-level classification problem that was introduced in Pang et al. (2002). The main result of their work was that they identified many small changes that improved document-level classification on product reviews. However they also introduced a review searcher that crawled search engine results to find reviews of a given product. This searcher classified each sentence in each review as positive or negative, which they used to score the product overall.

Dave et al. also found common instances of "the <x>", and treated <x> as an aspect of the product. They scored aspects by using the score given to the sentence it occurred in. They were thus able to display simple summaries of what reviewers thought of various aspects of products by displaying sentences containing references to those aspects. These summaries were further augmented with the average score given to all sentences containing references to each aspect.

Yi et al. (2003) made detecting aspects the focus of their work. They defined and evaluated three potential methods for extracting product features, largely based on POS tags and the definite Noun Phrases (NP). They evaluated each of these methods with two aspect selection methods, namely a likelihood test and a mixture model, and found that using a likelihood test with potential aspects needing to be definite, at the start of a sentence, and matching certain patterns of POS tags performed best. Next their system identified phrases containing the aspects that match a pattern database that they built from several linguistic resources. These patterns defined the scope of the sentiment as well as the tokens that indicated the polarity of the sentiment. For cases where the patterns matched, they made their final sentiment classification by checking the particular tokens against several sentiment lexicons. For cases where there was no match they looked for fragments containing the aspect and made the sentiment classification largely based on any modifying adjectives.

Later work by Hu and Liu (2004), was focused on taking aspect-based classifications and building them into a useful summary. To find aspects they used association mining with some pruning to find NPs that occurred frequently across their corpus, on the intuition that those NPs are likely to be aspects of the product. Next, they used a simple bootstrapping approach over WordNet to label adjectives as either positive or negative. They then used this list to find sentences that contained at least one aspect and one opinionated adjective. They could then classify the opinion the author has towards each aspect by taking a majority vote of the opinionated adjectives that appeared near each aspect, with ties broken by using the nearest adjective. The authors next proceeded to find infrequent aspects, which were not picked up by looking for common NPs. They found these by looking for NPs that contained opinionated adjectives, and added those NPs as aspects. In the final step they generated opinion summaries for each aspect by classifying each sentence referring to that aspect as positive or negative, based on the adjectives within it, as well as some extra modifications. The opinion summary is then a list of both the positive and negative sentences that refer to each aspect. Later work by Popescu and Etzioni (2005) improved on Hu and Liu's results, and provided a more detailed analysis of the stages of the system.

Kim and Hovy (2005) looked at extracting the pros and cons from online product reviews, with a specific goal of avoiding system output that merely notes the number of positive and negative reviews. For data they scraped epinions[3], a customer review website that allows users to list specific pros and cons separately to their main reviews. They used this data to train two ME classifiers, which they applied to unseen text-only reviews. The first classifier distinguishes between sentences containing irrelevant sentences and opinions while the second classifier determines whether the opinion in a sentence is a pro or con. For features they used unigrams, bigrams, trigrams, some positional features, and a list of pre-selected opinion-bearing words. Their system achieved 71% F-score in identifying opinions, and 61% F-score in classifying them as pro or con.

[3] http://www.epinions.com

While all of the above methods detect aspects based on some variation of term frequency, Titov and McDonald (2008) used topic modelling to cluster aspects, such that "bartender" and "waitress" might get clustered into a more generic "staff" cluster. The two methods that they examined were Latent Dirichlet Allocation (LDA) and Probabilistic Latent Semantic Analysis (PLSA). Their initial results showed that both of these methods tended to cluster terms into instances of specific products, rather than features of those products. To solve this they proposed to make LDA and PLSA "multi-grain", where there are topics generated from multiple levels of granularity. More specifically, they allowed a global distribution of topics as usual and a variable set of local topics. The variable local topics were found by instantiating several partially overlapping windowed distributions with coverage over the whole document. Then, for each word they sampled from one of the local windowed models or the global model, according to some distribution. They provided qualitative examples of the topics that their model produced, as well as a quantitative evaluation that showed that their method produced topics that outperformed a baseline on an aspect ranking problem.

There has also been work on jointly classifying the sentiment in a document at various levels. McDonald et al. (2007) proposed a structured model to jointly predict sentiment at both a sentence and document level. They showed that this approach yielded better classification accuracy than standard sentence models and sequential sentence models – similar to those proposed in Mao and Lebanon (2006) – for both levels of granularity. Nakagawa et al. (2010) proposed a similar joint modelling, except that they jointly modelled the sentiment of a sentence and the arcs in the dependency parse of the sentence. Their results showed an improvement over the standard bag-of-words model. While neither of these two approaches found the sentiment about aspects, both authors noted how this could be achieved.

Though Nakagawa et al. presented a model that relied on the dependency tree of sentences, they could not train from polarity-labelled tree fragments, as those labels did not exist. In recent work, Socher et al. (2013) presented a Sentiment Treebank, which they built from the movie review corpus introduced in Pang and Lee (2005). To build the treebank they first parsed the corpus with a standard Probabilistic Context-Free Grammar

(PCFG) parser. They then used Amazon's Mechanical Turk[4] to label all of the sentences and parse nodes in the corpus with a polarity score. This, along with the original annotations by Pang and Lee, gave them a polarity label for the document as a whole, every sentence, and every phrase node. Socher et al. then presented a novel variation on Recursive Neural Networks (RNN), which they call Recursive Neural Tensor Networks (RNTN). This novel variation is able to better capture the more subtle effects of negations and intensifiers, and it thus performs 5.4% better at the sentence-level classification task. This work also does not explicitly identify the aspects of products, however it provides a natural system to find sentiment about aspects, by using the scope of sentiments attached to an aspect through the annotated parse tree.

Somasundaran et al. (2008b) proposed to look at sentiment in a more discourse-aware manner by proposing *opinion frames*, which link pairs of opinions. In their work opinions can either be sentiment, which is positive or negative, or arguing, which is for or against, with either of these options having a target expression. Opinion frames then unite pairs of these opinions according to whether they have the same target, or whether they have targets that are alternatives to each other. Using this idea, Somasundaran et al. (2008a) produced a corpus and an annotation study. In later work, Somasundaran et al. (2009b) investigated whether detecting opinion frames could aid opinion polarity classification. They found that by incorporating discourse information they were able to improve opinion polarity classification recall, without harming precision. Later work (Somasundaran et al., 2009a) proposed to jointly identify opinions and opinion frames using graph-based methods. These ideas are described in more detail in Somasundaran (2010).

5.5 Opinion Mining in News and Politics

The work that we have discussed thus far has focused on product and movie reviews, where the communicative goal of the reviewer is to evaluate the item they are reviewing. This is a very natural setting for sentiment analysis, as these reviews will contain many opinionated statements, which will largely be about the item under review or attributes of that item. There are other areas, such as news and politics, where it is useful to know

[4] http://www.mturk.com

the opinions that people hold. However, these domains are not as straightforward as product reviews, as news text tends to cover multiple points of view, and political text tends to use the same language regardless of which side they support (Agrawal et al., 2003). Nonetheless, there has been a great deal of interest in extending opinion mining to these domains.

Kim and Hovy (2004) attempted to extract opinions about a given topic from news text, where they defined opinions as a tuple of a *topic*, *holder*, *claim*, and *sentiment*. Their method works by finding sentences that contain a reference to the topic, which is a given query, as well as at least one NE, which they considered to be the holder. Sentiment is then added to this tuple by running a classifier over the sentiment-bearing words that they identified with their own WordNet-based method. In later work, they extended their methods to new domains (Kim and Hovy, 2006a,b), such as online reviews.

Though reviews of movies and products have been the most popular domains for document-level sentiment analysis, political speech has also attracted attention. Thomas et al. (2006) created a corpus of U.S. congressional speeches about particular bills, where each speech was labelled with the congressperson's vote on the given bill. While these speeches could be classified using an SVM classifier with the familiar bag-of-words features, they found that they could achieve better accuracy by determining agreement between participants in the discussion. They did this by searching each speech for references to other congresspeople, and then using an SVM with windowed bag-of-words features to find whether the reference represented agreement or disagreement. The standard bag-of-words features achieved an accuracy of 66.05%, while the bag-of-words augmented with agreement links achieved 70.81%. Thomas et al. were able to avoid the issue of annotating political speech by using the actions of politicians as an indication of their opinions. However, these indications are not readily available for most forms of political discussion, as the speech itself is what informs listeners or readers about the opinions.

Yu et al. (2008) examined the differences in sentiment between congressional speeches, movie reviews, and news articles. Their work highlighted a number of key differences between the sentiment found in political speech and movie reviews, though their news article results were less informative for our purposes as they chose business articles. One of their findings is that congressional speeches are more opinionated than business news, but less opinionated than movie reviews. This matches intuition, as political speeches often include debates about what is factual, rather than the more straightforward opinions that are expressed in reviews. They also noted that the sentiment in movie reviews was often expressed through adjectives, while the congressional speeches tended to express opinions by discussing different topics, often in quite neutral tones. They argued that these findings indicate that searching for sentiment is insufficient for categorising political opinions, as many opinions are expressed without resorting to subjective language. This has proved to be a challenge for text-based opinion mining techniques.

5.5.1 Stance Classification

Agrawal et al. (2003) proposed the task of clustering users on a debate forum into opposing camps, a task which is often called *stance classification*. They noted that one of the challenges in doing this is that the words and phrases used by both sides of politics are largely similar. This makes the standard bag-of-words feature set less effective, as n-grams alone are less discriminative. To resolve this issue Agrawal et al. noted that users tend to quote other users with whom they disagree, which provides an alternative way to find the stance of the discussants. Agrawal et al. exploited this link structure by finding the maximum cut partition of the citation graph, which ensures that the maximum number of links cross the partition. This purely link-based approach outperformed text-based approaches by a wide margin.

While Adamic and Glance (2005) did not classify political opinions *per se*, they did analyse the link structure between liberal and convervative blogs in the lead up to the 2004 U.S. presidential election. They followed 20 liberal and 20 conservative blogs for a period of three months, and found that both liberal and conservative bloggers tended to cite blogs that were on the same side of politics as them, with neither side tending to cite blogs from the other side of politics. This finding for blogs is opposite to the findings from Agrawal et al. (2003), who found that in discussion forums, users tended to cite other users that they disagreed with. The authors were also interested in analysing whether there was an “echo chamber” effect, where bloggers would echo the arguments and links of other bloggers with whom they share a political orientation. To study this they looked at both the links between blogs and frequent phrases from each blog as compared to background weblog data. Interestingly, although conservatives tended to have a denser link structure within their community, neither side exhibited a stronger echo chamber effect than the other.

Efron (2004) also noted the phenomenon of political blogs citing other blogs that they agree with, rather than blogs that they oppose. Stance classification on these blogs could be performed using an adaptation of Agrawal et al's approach, where the min-cut algorithm would replace the max-cut algorithm. Efron takes a different approach, by employing *cocitation analysis*, which is an adaptation of Turney's PMI-IR, to use the given link structure, rather than the link structure from a search engine. For this method to work a small number of documents that exhibit a clear ideological stance must be labelled, but once these documents have been annotated, any other document in the link graph can be classified using this method. Using this approach Efron was able to achieve an accuracy of 96% in classifying the ideological stance of blogs.

Though Efron's work presented an interesting method, the evaluation was relatively small. Mullen and Malouf (2006), and later Malouf and Mullen (2008), addressed this by presenting a new dataset created from over 75,000 comments on a political debate website. They attempted some classification experiments based on the text of the posts and found that the performance is disappointing, leading them to agree with both Efron and Agrawal et al. that the citation structure would provide a better method for classification. In fact, their experiments showed that classifying users as *left* or *right* by assigning them to the label opposite that of the people they cite yields better performance than any of the text-based methods, even though only 55.7% of users cite other users.

Somasundaran and Wiebe (2010) extended their previous work on products (Somasundaran and Wiebe, 2009), by classifying the stances of users in political debates without using a linked graph. They noted that using raw sentiment is insufficient for this task, as discussants argue about what is true, and may use both positive and negative terms to describe aspects of either side of the debate. To remedy this they developed arguing features that are largely based on an argument lexicon that they constructed. They evaluated their work on a corpus of political debates from an online forum, and found that the arguing features along with features based on a sentiment lexicon outperform simple unigrams, arguing features alone, and sentiment features alone. In later work, Anand et al. (2011) showed that it is particularly difficult to classify posts that rebut the point of a previous poster, even for humans.

The work discussed so far has looked at political texts that are either congressional speeches, which are reasonably long, or forum posts, which have a reasonably dense link structure that aids classification. Several works (Rao et al., 2010; Pennacchiotti and Popescu, 2011) have instead looked at classifying the political stance of Twitter users, which presents new challenges as the text is often malformed. Similarly to the work on blogs and speeches, these approaches primarily rely on the graph structure resulting from mentions of other users, retweets, and hashtags. Though these studies showed that classifying the political stance of Twitter users was feasible, other studies (Metaxas et al., 2011; Gayo-Avello et al., 2011) showed that predicting electoral results using Twitter is not better than chance.

In an interesting variation on stance classification, Park et al. (2011) defined debates according to opposing parties, and classified news articles according to the side they broadly agree with. They quoted an example of news coverage of a proposed health care bill, where one news article discusses increased coverage of the population, while another discusses the increased cost of insurance premiums. They argued that each of these articles exhibits a preference for a side of the debate, which can be predicted. While directly annotating the side produced poor agreement, they showed that defining primary participants in the debate and annotating whether an article supports or opposes these participants increases annotator agreement substantially. Using this formulation, they created a corpus of 250 articles covering fourteen different contentious issues. Though their main task was to classify whole articles, they showed that they could extract and partition disputants with a precision of about 70% and a slightly lower recall. In evaluating article classification, they compared a similarity-based clustering method, a quotes-based method that classifies articles according to which sides' disputants are quoted,[5] and their main disputant relations method. Their results showed that the disputant-based approach yielded the best results for seven out of fourteen topics, the quotes-based method yielded the best result for five topics, and the quotes and disputant methods tied for two topics.

Not all topics of debate have two sides, as the papers discussed thus far have assumed. Recent work (Abu-Jbara et al., 2012; Hassan et al., 2012; Abu-Jbara and Radev, 2012) has taken an unsupervised approach to clustering participants in online debates into subgroups, where the number of subgroups is not necessarily fixed a priori. Abu-Jbara et al. (2012) performed this task by creating an attitude vector for each discussant, which scores their opinions towards other discussants and other entities in the text. They then used clustering techniques to find subgroups in the resulting vector space. By contrast, Hassan et al. (2012) looked at just the relations between discussants, which they used to form a signed graph that they divided into subgroups using a graph-based optimisation technique.

One of the challenges in stance classification is that the evaluation data that methods have used has been drawn from different sources and with different underlying assumptions. The work of Walker et al. (2012) aimed to fix this by presenting the Internet Argument Corpus (IAC), which is a large-scale collection of posts from debate forums. The corpus contains 130,206 posts that were manually labelled by the discussants with the topic and side that the poster supports. Some subsets of this were also labelled with other relevant information.

5.6 Opinion Question Answering

In traditional QA, a user would ask a question that requires a fact as an answer, such as "when was Barack Obama first elected president?" In opinion question answering, the questions may require opinions as answers, such as "was Barack Obama an effective president?" In other formulations, the user might ask a factual question that requires

[5] Quote extraction and attribution is not the focus of their work, and, as such, they provide limited details of their method and no specific results, so their work was excluded from our literature surveys in Chapters 2 and 3.

knowledge of opinions to answer, such as “who thinks Barack Obama was an effective president?”

Yu and Hatzivassiloglou (2003) explicitly targeted their work towards opinion question answering, although they did not attempt it as a task. They identified some of the key components that would be required for such a system and presented results for those components. The first part is document classification of news articles, where the aim is to separate editorials and opinions from regular news reports. For this they used a simple NB classifier with bag-of-words features, which achieved an F-score of 97%.

This gave the authors the ability to separate opinionated documents from factual ones, but they noted that for opinion question answering finer-grained results are required. As such, the next part of their work focused on finding subjective sentences and determining their polarity. In order to do this Yu and Hatzivassiloglou annotated 400 sentences from their main corpus, with agreement ranging from 0% to 77% depending on the label. For identifying subjective sentences they found that a NB classifier with lexical features achieved 91% precision, while their sentence-level polarity classifier achieved 90% accuracy.

Yu and Hatzivassiloglou’s sentence-level annotations demonstrated that achieving good annotator agreement on fine-grained annotations of opinion can be difficult. Work by Wilson and Wiebe (2003), Wiebe et al. (2003) and later Wiebe et al. (2005) investigated this problem and established a large-scale corpus, known as the Multi-Perspective Question Answering (MPQA) corpus. The MPQA contains detailed annotations of opinions in news articles, with annotations of subjective sentences, speech events, direct subjective expressions, sentiment expressions, agents, and targets. The goal of these annotations was to mark out individual expressions of opinions and sentiment, which could then be avoided or targeted by an opinion question answering system. Later work (Wilson et al., 2005a,b; Choi et al., 2006) proposed systems that can recreate these annotations on unseen documents.

Stoyanov et al. (2004) were the first to build a corpus that included questions requiring opinions as answers. The corpus, called OPQA, was built from 98 of the documents that are in the MPQA corpus, with the 98 documents being split amongst four topics. The annotators created 30 questions, with half requiring factual answers and half requiring opinionated answers, and then annotated every answer and partial answer found in the 98 documents. Stoyanov et al. (2005) used this corpus to identify four main challenges in identifying opinionated answers:

1. Opinionated answers are approximately twice as long as factual answers
2. Opinionated answers are much more likely to be only partial answers
3. Opinionated answers are more syntactically varied
4. Opinionated answers are approximately half as likely to correspond to a single syntactic type

Based on these challenges, Stoyanov et al. argued that MPQA is more difficult than standard QA, and would require extra processing. The final contribution of Stoyanov et al.'s work is that they built a prototype MPQA system. They showed that traditional QA subsystems on their own do not perform well. However, they can be improved by using subjectivity classifiers that exclude sentences (and potential answers) that contain only facts. They further experimented with finding sentences that contain an opinion with a source, however this yielded mixed results.

Following on from Stoyanov et al., the MPQA task was included as part of the Text Analysis Conference (TAC) 2008 Question Answering Track (Dang and Owczarzak, 2008). The competition attracted nine entrants and involved two types of questions, *rigid list* questions, where systems had to list entities that held a certain opinion, and *squishy list* questions, where the systems had to provide lists of opinions as answers. The rigid list questions were evaluated using F-score, while the squishy list questions were evaluated using pyramid F-score (Dang and Lin, 2007). Table 5.1 shows an example of each type of question along with some example answers.

The THU QUANTA (Li et al., 2008) system achieved the best combined score of 16.8%, compared to the human baseline of 50.5%. They used four query generation methods for document retrieval and then removed documents that did not match the sentiment of the question, as determined by a simple polarity lexicon. They then extracted appropriate

Rigid list	Squishy list
Who likes Trader Joe's?	**Why do people like Trader Joe's?**
BLOG06-4201 Peggy Archer	BLOG06-3227 Trader Joes is your destination if you prefer Industrial wines (unlike Whole Foods).
BLOG06-5961 david Ford	BLOG06-2494 Everytime I walk into a Trader Joes it's a fun filled experience, and I always learn something new…
BLOG06-5961 Michelle	BLOG06-4400 Sure, we have our natural food stores, but they are expensive and don't have the variety that Trader Joe's has.

Table 5.1: Examples of rigid list and squishy list questions and answers. The answers include a reference to the document they came from. Examples are from Dang and Owczarzak (2008), with all text in its original form.

snippets with a combined topic relevance and sentiment match score. The main contribution of their work was in answer extraction, where they further subdivided the types of questions based on the type of answer required, with each subtype having its own answer extraction component.

Though the THU QUANTA system achieved the best result on the combined and rigid list scores, the IIT Hyderabad system (Varma et al., 2008) achieved the best result on the squishy list questions. Their overall architecture was broadly similar to traditional QA systems, however their answer ranker employed a combination of features based on in-context and out-of-context topic relevance, a subjectivity score from a subjectivity classifier, and a question polarity agreement score from a polarity classifier. Their pyramid F-score for squishy list questions was 0.1864, compared to 44.6% for the human baseline and 17.3% for the next nearest team.

Overall, the results of the competition show that opinion question answering requires more research before it can be used effectively.

5.7 Summary

In this chapter we have surveyed some of the key tasks in opinion mining and sentiment analysis, as well as some of the key approaches to these tasks. For the majority of the literature, which is focused on classifying the opinions found in reviews, there is an assumption that each document has a single source, i.e. the document's author, whose communicative goal is to evaluate a well-defined target, which is the given product or movie. This assumption is well-founded for many applications, as it is rare to discuss unrelated items in a review, unless it is to compare the item under review with a competing item. Some work on product reviews goes further and seeks to identify aspects of the products that the reviewer has discussed. In some political texts, these assumptions hold as well. Congressional speeches are typically about a particular bill, and as Thomas et al. (2006) showed, we can use the congressperson's actual vote to determine whether or not they support the bill in question. Blogs also tend to exhibit a political preference, and when they discuss political topics, bloggers will typically include their own opinions.

News on the other hand has proved more difficult. While Park et al. (2011) were able to classify whole news articles according to the political side the article is most sympathetic to, in general news text is fairly objective. As such, the assumption that the author is evaluating some well-defined target does not hold for news. Instead, journalists will present a range of viewpoints from other sources, which need to be detected. These viewpoints also need to be categorised in some way, as there is no easy label given by the authors themselves. It is these issues which lead us to define our opinion annotation scheme, as well as a corpus and some preliminary results in the next chapter.

6 · Opinion Corpus

In the previous chapter we gave an overview of the various opinion related tasks in NLP. This chapter covers the definition of a new task, as well as an annotated corpus and some pilot experiments. Our main interest in establishing this task is in extracting opinions from news articles, as the opinions are often by salient entities about important events. As we have argued in previous chapters, these opinions are most often found in quotes, as journalists can use quotes to provide evidence of an opinion, rather than relying on an assertion.

In sentiment analysis over movie or product reviews, polarity labels are commonly used, as the target of the polarity is clearly the item under review. In political debates, however, polarity labels do not as obviously apply. Consider the topic of climate change. Labelling a quote as either positive or negative about climate change raises the question of what exactly positive and negative mean in this context. A naïve interpretation would be that the speaker is literally in favour of the Earth's climate changing. However, a reasonable person would find this interpretation unlikely. A negative polarity is even more difficult, as it could mean that the speaker is negative about changing the climate, or about the scientific case for climate change, or about some other aspect of the topic, such as a particular mechanism for addressing the problem.

The granularity of the opinions expressed is another factor that can make opinion mining in this genre difficult. When the data under consideration is a review, the granularity of the opinions is largely defined by the intended application. For example if the goal is to assign a meta-score to a product, the appropriate granularity is document level. If, on the other hand, the application is to highlight the good and bad features of a product, then the granularity should be set at aspect level.

These considerations do not apply as well to news documents. A document-level sentiment classification of a news document is in most cases meaningless, as an unbiased article should present multiple points of view. On the other hand, breaking the opinions down into their constituent parts risks losing the overall point of view that an entity is trying to express. The following real-world example illustrates this:

> "I am pro-life and also pro-choice and I don't find any conflict in that."

If we were to break the example down into constituent parts we may find that we have two opinions being expressed, one is that the speaker is pro-life while the other is that the speaker is pro-choice. However when we look at the quote as a whole, it is clear that the speaker is objecting to the choice of labels the two principal sides of the debate have chosen for themselves.

Issues with granularity and the meaning of polarity affect all opinion-related tasks. In our work we avoid making a single decision about granularity and polarity by using *position statements*. We define a position statement as a clear statement of a viewpoint on a particular topic, which aligns with a side of the debate. Quotes related to this topic can then be labelled as *supporting*, *neutral* towards, or *opposing* the given position statement. While this does not solve the broader issues with granularity and polarity in opinion mining, it does allow us to disambiguate the meaning of polarity and clarify the granularity for the specific issue referred to by the position statement.

With this formulation we construct a corpus covering seven topics, with 100 documents per topic, for a total of 2,228 quotes. All quotes in our corpus were annotated by three annotators, with 99% of the quotes having at least two-way agreement. Most disagreement occurred between each of the polar labels and neutral, which indicates that annotators had difficulty distinguishing when opinions had become relevant or clear enough to constitute support or opposition to the position statement. We also found that despite the evidential role that quotes play in news, the context they are found in tends to alter their interpretation towards being more polar.

We present pilot experiments for the task of finding whether a quote is polar or not polar. These experiments use a state-of-the-art subjectivity detection system, along with a simple learning technique to classify quotes. The method achieves an F-score of between 37% and 50% for detecting polar quotes.

This chapter first describes the corpus and annotation process in more detail in Section 6.1. As this is a challenging annotation task we next discuss some of the key difficulties faced by annotators in Section 6.2, before presenting and discussing the pilot experiments in Section 6.3.

This chapter is based on work that was published in the Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics.

6.1 Corpus

A major contribution of this work is that we construct a large corpus for this task. To build this corpus we employed three annotators, one of whom is the author, while the other two were hired using the outsourcing website Freelancer.[1] In this section, we describe the data collection, preprocessing steps, the annotation scheme, and the label distributions in the corpus.

6.1.1 Position Statements

Our goal in this study is to determine which side of a debate a given quote supports. Existing opinion labelling schemes do not work here for several reasons, which we will illustrate with the following three examples:

(13) a. **Supports:** "I now believe that the time has come. . . for us to have a truly Australian constitutional head of state."

b. **Neutral:** "The establishment of an Australian republic is essentially a symbolic change, with the main arguments, for and against, turning on national identity. . ."

c. **Opposes:** "I personally think that the monarchy is a tradition which we want to keep."

[1] http://www.freelancer.com

In Example 13a, the speaker is making a clear statement about their preference for a republic over the status quo. Despite this clear statement it is difficult for existing labelling schemes to classify the example. Annotating the whole example, or its constituent tokens, with sentiment polarity is unenlightening as the statement is fairly neutral. Labelling the example as subjective or objective is also somewhat challenging as the speaker is making an objective statement about their own views on a subjective topic (a problem shared by Example 13c).

Using an expression-based scheme, such as the one defined in Wiebe et al. (2005), does make sense for extracting some opinions, but has its own issues. In Example 13a, the topic we are interested in, i.e. whether or not Australia should become a republic, is only referenced indirectly via the phrase a truly Australian constitutional head of state. The other two examples do contain more explicit references to the topic, but they use different labels, i.e. an Australian republic and the monarchy. While it might seem obvious that talking positively about an Australian republic or the monarchy would equate to support for that concept, it is not as clear that talking negatively about either of these textually-anchored targets is equivalent to opposition. In order to answer questions like these we need a corpus that labels these examples with more direct notions of support or opposition to a topic.

Directly labelling quotes as supporting or opposing a simple topic descriptor has similar problems. For example labelling quotes as supporting or opposing a descriptor such as abortion seems overly simplistic. While the *opposing* label is reasonably interpretable, the *supporting* label could mean many different things. Furthermore the descriptor itself can be an issue, as the term describing the debate, such as abortion, may not necessarily be a label for what the debate is about, which in this case is who has the right to decide whether or not to carry out the procedure.

An alternative approach might be to instead label the quotes with the labels that the sides of the debate use, which in this case would be pro-choice and pro-life. The problem with this is that those labels might mean different things to different people, and for topics without such handy labels we would need to define them along with a description of what they mean. This description would be close to what we are proposing here.

As a result of these issues, we propose *position statements*, which are largely unambiguous statements of a viewpoint on a given topic. Quotes can be labelled as *supporting*, *neutral* towards, or *opposing* a given position statement. We found in early iterations of our annotation scheme that these statements greatly clarified the meaning of the labels, and resulted in much better inter-annotator agreement. This means that we are able to characterise the opinions in news articles at the level that we are actually interested in. By using position statements we are also able to be as precise as we can about the topic in question, and for more complex topics we could define multiple statements, though in this work we only consider one per topic.

Using position statements also makes our work somewhat comparable to work on debate summarisation and subgroup detection (Somasundaran and Wiebe, 2010; Abu-Jbara et al., 2012; Hassan et al., 2012). Authors attempting these tasks have often used data from online debate forums. Before posting about a topic, users of some of these forums need to participate in a poll which forces them to choose from two or more propositions that represent the major viewpoints on the topic.

Our work applies a similar idea to the news domain, where the discussants are the individuals who have been quoted, and the position statements are the propositions which their opinions are evaluated against. To a lesser extent, our work can be considered related to the work on product and movie reviews if we consider sentiment analysis in reviews to be an evaluation of the review against the implicit position statement that the product or movie under review is good.

The idea of position statements is also related to the idea of perspectives. Work by Beigman Klebanov et al. (2010) looks at perspectives on four different levels, including a low-level idea of perspective that aligns with our idea of opinions. If we view position statements as a level of perspective, then it's clear that they act to increase the level of specificity when compared to simple polarity labels.

Figure 6.1 shows the position statements that we use in this work. For six of the seven topics we have attached the *supports* label to the side that supports change from the status quo, with the remaining topic, *abortion*, attached arbitrarily as the articles cover multiple

Abortion: Women should have the right to choose an abortion.

Carbon tax: Australia should introduce a tax on carbon or an emissions trading scheme to combat global warming.

Immigration: Immigration into Australia should be maintained or increased because its benefits outweigh any negatives.

Reconciliation: The Australian government should formally apologise to the Aboriginal people for past injustices.

Republic: Australia should cease to be a monarchy with the Queen as head of state and become a republic with an Australian head of state.

Same-sex marriage: Same-sex couples should have the right to attain the legal state of marriage as it is for heterosexual couples.

WorkChoices: Australia should introduce WorkChoices[2] to give employers more control over wages and conditions.

Figure 6.1: Topics and their position statements.

jurisdictions with different laws. This has made the *supports* label align with the left-wing position for six of the seven topics.

6.1.2 Data Collection and Preprocessing

Our data is drawn from a subset of the Sydney Morning Herald (SMH)[3] archive, which ranges from 1986 until 2009. Our corpus covers seven topics that were subject to debate within Australian news media during the time covered by the archive. For each of the topics we created a search query and used Apache Solr[4] to find the top 100 documents that matched the query. The topics covered, the Solr queries used, and the position statements are shown in Table 6.1.

For preprocessing, each document was first tokenised and POS tagged. We then ran NER and NEL (Hachey et al., 2013) to discover the entities in the document and disambiguate them back to a KB. While NEL is not strictly necessary for the task as defined in the previous section, it is likely to be useful for applications of this data.

This corpus was created before we had finished the full suite of quote extraction and attribution tools described in Chapters 2, 3, and 4. As a result we only extract direct quotes

[3] http://www.smh.com.au

[4] http://lucene.apache.org/solr/

Topic	Search query
Abortion	`"abortion" "pro choice" "pro-choice" "pro-life" "pro life"`
Carbon tax	`"carbon tax"`
Immigration	`"immigration" "migrant workers" "migrant labor" "foreign workers"-"illegal"`
Reconciliation	`+"reconciliation"+("sorry" "apology") "stolen generation"`
Republic	`"republic" "monarchy"`
Same-sex marriage	`"same-sex marriage" "gay marriage" "same sex marriage" "single-sex marriage" "single sex marriage" "homosexual marriage" "lesbian marriage"`
WorkChoices	`+"work choices" -"fair work" +timestamp: [2006-01-01 TO 2007-04-01]`

Table 6.1: Topics and the search queries used to find relevant documents.

and the directly-quoted portion of mixed quotes. Annotators were asked to consider the indirect portions of mixed quotes as part of the quote, as too many mixed quotes were meaningless without the indirect portion. Quotes were attributed to their speakers using the version of the binary class-labelled attribution system using no sequence features from O'Keefe et al. (2012).

6.1.3 Annotation Scheme

Our annotation scheme was designed for both expert and non-expert annotators, and was iteratively refined based on pilot annotations. For our final scheme annotators were asked to follow a series of steps:

1. Check that the quote span and the speaker are correct
2. Check that the quote is on topic and that it is actually a quote
3. Does the quote provide evidence for the speaker's position on the topic, ignoring all context and outside knowledge?
4. What if you consider the context that the quote appears in?

For the first step annotators were asked to mark cases where the upstream processes had incorrectly identified either the speaker or the span of the quote, although for these cases they were asked to continue annotating as though the system had been correct. For step two, annotators were provided with options to indicate if the quote was *off-topic* or *not a quote*. Annotators were encouraged to prefer *neutral* to *off-topic*. The *not a quote* option covers things like book titles, excerpts from reports, scare quotes and similar items that are not quotes according to the definition in this thesis. If either of these options were selected, they were asked to skip the remaining steps and move to the next quote.

For step three, annotators were asked to interpret the opinions in each quote without considering any of the context of the quote. In other words they were asked to only use the text of the quote itself as evidence for an opinion, not the speaker's prior opinions or the text of the document. Though we are principally interested in quotes as evidence for opinions, we are also interested in analysing the effect of context on the interpretation of each quote. In order to analyse this, in step four we asked annotators to consider the text surrounding the quote while annotating, although they were still asked to ignore the prior opinions of the speaker.

Throughout the annotation process, the interface displayed the position statement underneath the quote being annotated. For both steps three and four, the annotators were asked to determine whether the quote they were considering provided evidence of *strong or clear opposition*, *opposition*, *neutrality*, *support*, or *strong or clear support* for the position statement. Appendix B contains the full annotation scheme.

6.1.4 Invalid Quotes

In order to minimise the noise in our corpus, we opted to discard quotes that any annotator had marked as being invalid. From the full set of 3,428 quotes, 1,183 (35%) quotes were removed, leaving 2,245 (65%) quotes. Table 6.2 shows the per-topic breakdown of the reasons quotes were marked invalid. The wrong span and no sensible choice options accounted for 1% or fewer for each topic, so were excluded from the table. The single largest reason was that they were off-topic, which accounted for 23% of the quotes in the corpus. As these quotes make up such a large portion of the corpus, it is likely that in order to label opinions in news, a system would first have to identify which parts of the

Topic	Quotes	Valid (%)	Not a quote (%)	Off-topic (%)	**Total invalid (%)**
Abortion	515	67	12	28	33
Carbon tax	416	67	17	18	33
Immigration	509	49	12	43	51
Reconciliation	644	80	15	5	20
Republic	458	76	16	10	24
Same-sex marriage	466	53	24	32	47
WorkChoices	420	64	15	31	36
Total	3,428	65	16	23	35

Table 6.2: Number of quotes annotated, and the percentages that were invalid for various reasons. Note that annotators could select multiple reasons for a quote being invalid.

text were relevant to the topic in question, rather than considering whole articles as being on topic.

The annotators further indicated that 16% were not quotes according to our definition, which was largely made up of references to other articles and excerpts from reports. Better pre-processing would have removed the majority of these cases, as title based references to other articles are likely to be easy to disqualify, as are quotes attributed to organisations (assuming accurate attribution). There were also a small number of cases where the quote span was incorrect, which was due to missing quotation marks.

6.1.5 Agreement

Table 6.3 shows both Fleiss' κ and the raw agreement averaged between annotators for each topic. To calculate these numbers we collapsed the two supporting labels together, as well as the two opposing labels, such that we ended up with a classification of *opposes* vs. *neutral* vs. *supports*. In terms of raw agreement the no context and context cases came to 0.69 and 0.66 respectively, while the κ values were 0.43 and 0.45, which is in the moderate agreement range. We consider this to be a reasonable result, given the complexity of the task.

Topic	Quotes	No context		Context	
		AA (%)	κ (%)	AA (%)	κ (%)
Abortion	343	77	57	73	53
Carbon tax	278	71	42	57	34
Immigration	249	58	18	58	25
Reconciliation	513	66	37	68	44
Republic	347	68	51	71	58
Same-sex marriage	246	72	51	71	55
WorkChoices	269	72	45	65	44
Total	2,245	69	43	66	45

Table 6.3: Average Agreement (AA) and Fleiss' κ over the valid quotes

Topic	Quotes	No context		Context	
		AA (%)	κ (%)	AA (%)	κ (%)
Abortion	343	78	52	74	46
Carbon tax	278	72	39	59	19
Immigration	249	58	8	58	14
Reconciliation	513	66	31	69	36
Republic	347	69	39	72	41
Same-sex marriage	246	73	43	73	40
WorkChoices	269	73	40	67	32
Total	2,245	70	36	68	32

Table 6.4: Average Agreement (AA) and Fleiss' κ when the labels are neutral versus non-neutral

Intuitively we might expect that there is much greater confusion between neutral and either of the two polar labels, than there is between the two polar labels themselves. To examine this, we merged all of the non-neutral labels into one group and calculated the inter-annotator agreement between the non-neutral group and the neutral label, as shown in Table 6.4. For the non-neutral vs. neutral agreement, we find that despite stability in the raw agreement, Fleiss' κ drops substantially, to 0.36 and 0.32 for the no context and context cases respectively.

For comparison we remove all neutral annotations and focus only on disagreement about the polarity of labels. For this result we cannot use Fleiss' κ directly, as it requires a fixed number of annotations per quote, however we can average the pairwise κ values between annotators, which results in values of 0.93 (no context) and 0.92 (context). Though they are not directly comparable, the magnitude of the difference between the neutral vs. polar numbers of 0.36 and 0.32, and the polar confusion numbers of 0.93 and 0.92, indicates that deciding when an opinion is sufficient evidence of support or opposition is clearly the main challenge facing annotators.

Differentiating between a neutral and polar quote was clearly the biggest challenge facing our annotators. However, our annotators also reported that it was more difficult to identify support for the position statement than it was to identify opposition to it. The core reason for this difficulty was that supporting the position statement tended to mean supporting a quite specific form of change. The most notable example of this is the *carbon tax* topic, where annotators were asked to annotate whether a speaker supported a carbon tax to mitigate the effects of climate change. Annotators found it difficult to judge whether speakers expressing support for alternative measures, such as funding for renewable energy projects, implied support for a carbon tax. By contrast, on the opposing side of the debate, speakers would often either express opposition to a carbon tax specifically, or they would express opposition to any measures at all. This made it easier for annotators to identify cases of opposition. Despite this anecdotal difficulty, there was no significant difference in the agreement statistics.

Topic	Quotes	No context			Context		
		3-way	2-way	None	3-way	2-way	None
Abortion	343	64	36	0	57	42	<1
Carbon tax	278	57	41	2	38	60	2
Immigration	249	35	64	1	35	63	1
Reconciliation	513	49	51	1	53	46	1
Republic	347	54	46	1	59	46	1
Same-sex Marriage	246	55	45	0	56	43	1
WorkChoices	269	59	39	1	46	53	1
Total	2,245	53	46	1	50	49	1

Table 6.5: Agreement (%) between annotators over the valid quotes.

6.1.6 Label Distribution

Table 6.5 shows the proportion of cases of three-way agreement, two-way agreement, and three-way disagreement in the corpus. To adjudicate the annotators decisions, we opted to take a majority vote for cases of two-way or three-way agreement, while discarding cases where annotators could not agree at all. For the corpus as a whole 99% of the quotes had either two-way or three-way agreement, and for the no context and context cases there were only 17 and 33 cases of complete disagreement respectively

The distribution of adjudicated labels in the corpus is shown in Table 6.6. For both the no context and context cases the largest class was neutral with 61% and 46% of the corpus respectively. This result shows that even though direct quotes are largely used as evidence, their interpretation can be altered by the context the journalist places them in. We speculate that this difference is largely due to anaphoric expressions within the quotes getting resolved to the journalist's own text.

In terms of the polar quotes, the opposing class was generally about half as common as the supporting class, with only the *WorkChoices* topic showing slightly more quotes opposed than in favour. Since the SMH is generally considered a left-leaning news provider, this is most likely caused by our position statements aligning support with a broadly left-wing position for all topics except *WorkChoices*. There may also be an inherent bias towards the *support* label in the set up of our task.

Topic	No context				Context			
	Quotes	Opp.	Neut.	Supp.	Quotes	Opp.	Neut.	Supp.
Abortion	343	13	63	25	340	16	52	32
Carbon tax	273	9	70	21	273	14	44	42
Immigration	247	9	72	19	245	12	64	23
Reconciliation	509	5	57	38	503	7	42	50
Republic	345	24	48	28	342	32	37	32
Same-sex marriage	246	16	55	28	243	25	38	37
WorkChoices	265	14	72	14	266	26	50	24
Total	2,228	12	61	26	2,212	18	46	36

Table 6.6: Label distribution in percentages for the final corpus. The discrepancy between the number of quotes for the no context and context cases comes from discarding cases of three-way disagreements.

6.2 Observed Annotation Challenges

Most of the quotes in our corpus do not make clear and unambiguous statements about the topic of the article, such as: "we have to completely outlaw all abortions." Much more frequently they will contain a more complex opinion that is difficult to label. In refining our annotation scheme and building our corpus we noted several common factors that make annotation difficult.

6.2.1 Opinion Relevance

When discussing a particular topic, journalists will often delve into the various aspects and related opinions that people hold on that topic. This is challenging, as annotators need to decide whether a particular quote is on-topic enough to be annotated. Let us consider two quotes on the *carbon tax* topic. The first is a clear and unambiguous statement of an opinion:

> "Whether it's a stealth tax, the emissions trading scheme, whether it's an upfront and straightforward tax like a carbon tax, there will not be any new taxes as part of the Coalition's policy"

In the same document, however, is another quote by the same speaker:

> "I don't think it's something that we should rush into. But certainly I'm happy to see a debate about the nuclear option."

In the first quote the speaker is clearly voicing opposition to a tax on carbon, which is easy to annotate with our scheme. However in the second quote, the speaker is discussing nuclear power as an alternative to a carbon tax. In this case it is much more difficult for an annotator to decide whether the quote should be marked as *off-topic* or *neutral*.

While confusion between *neutral* and *off-topic* is more common, there are also situations where it is difficult to decide between a polar option and *off-topic*. For example, a common occurrence in the carbon tax topic was for a speaker to call for action on climate change, without specifying exactly what that action should be. In those cases there is some evidence that the speaker would support a carbon tax, as a tax on carbon constitutes action on climate change, however without an explicit statement about a carbon tax it remains a somewhat difficult choice for annotators to make.

6.2.2 Obfuscation and Self-contradiction

While journalists will usually quote someone to provide evidence of the person's opinion on a topic, there are some cases where journalists include quotes to show that the person is inconsistent or self-contradictory. The following quotes by Alexander Downer (a conservative Australian politician) were both included in a single article, in order to illustrate that his position on the republic debate was unclear and potentially not consistent.

> "My point is that. . . the most potent argument in favour of the republic, is that why should we have a Briton as the Queen – who, of course, in reality is also the Queen of Australia – but a Briton as the head of State of Australia"

> "The Coalition supports the Constitution not because we support the. . . notion of the monarchy, but because we support the way our present Constitution works"

The above example also indicates a level of obfuscation that is reasonably common for politicians. Neither of the quotes actually expresses a clear statement of how the speaker feels about a potential republic. The first quote is an opinion about the strongest argument in favour of a republic, without necessarily making that argument, while the second quote states a party line, with a caveat that might indicate a personal disagreement with that party line. These statements are often difficult for annotators to label consistently.

6.2.3 Annotator Bias

Our annotation task requires an annotator to judge what constitutes support or opposition to a certain statement. This task, much more than other NLP tasks, is very prone to influence from the annotator's own biases. In particular, an annotator may be influenced by their own political or cultural background, their own opinion about the topic or speaker, or their level of knowledge about the topic or speaker. While position statements make the task much clearer than simple polarity labels, there is still considerable room for error.

6.3 Pilot Experiments

In this section, we present pilot experiments for the task of identifying which quotes are labelled with an opinion. These results use OpinionFinder (Wilson et al., 2005a), which is a state-of-the-art subjectivity detection and opinion classification system. We leave building a system capable of performing the full task to future work.

6.3.1 Approach

We expect that quotes that have been marked as providing evidence of an opinion will generally be more subjective. As such, we have opted to use the subjectivity labels produced by OpinionFinder to classify quotes as polar versus not polar. These labels are produced on a per-sentence basis, so to get a per-quote classification we removed all non-quoted tokens from each document and treated each quote as a sentence. OpinionFinder produces these labels using the Naïve Bayes (NB) classifier from Wiebe and Riloff (2005). Training data for the classifier is built using two high precision rule-based methods. The predictions of these classifiers are then used to train the NB classifier. As the rule-based classifications would potentially not capture the full range of sentences, the authors then introduce a self-training step, where the NB classifier is used to classify the training data. The top 50% of its predictions are then used to train the final NB model. For features the classifier uses a set of clues that were used in the rule-based classifiers, extraction patterns from Riloff and Wiebe (2003), and features for certain POS tags. The retrained classifier achieves an accuracy of 73.4% in classifying sentences as subjective versus objective.

We present two sets of results that both rely on OPINIONFINDER. The first set simply uses the labels that OPINIONFINDER produces, with no further processing. OPINIONFINDER produces both a label and a confidence value. We can use these, along with a small amount of labelled data, to find a more optimal decision boundary for this task. We take ten of the labelled documents from each topic and use these as training data. The confidence score for quotes marked subjective is left as is, while the score for quotes marked objective is negated, so that we get a score for each quote in the range of $[-1, 1]$. Quotes that meet the minimum confidence threshold are considered to be polar, while quotes below the confidence threshold are considered to be neutral or invalid. We find the boundary by sorting the quotes based on their scores, then testing the values in between each consecutive pair of quotes. We use the value that yields the highest F-score for the polar label on the training data.

We run separate experiments for each of the topics, where we compare the label provided by OPINIONFINDER with the learned boundary. For each topic, we run seven-fold cross validation with ten documents used as training data, while the remaining 60 are used as test data. This is an unusual setup, as it is more common to use the larger set as training data, rather than as test data. The reason for this approach is that we consider a small amount of training data to be a requirement for any new topic, so we want to evaluate how well the systems do with this small amount of data.

6.3.2 Results

Tables 6.7 and 6.8 show the results of our pilot experiments in distinguishing polar quotes from non-polar quotes. The results of the experiments show that we can achieve an F-score for the polar class of between 22% and 63%. The classifier with the learned boundary achieved a higher F-score for five of the seven topics, which was largely driven by higher recall with slightly lower precision. In fact, for the learned boundary, the difference in precision and recall ranged from 31% to 48%. We investigated this further and found that even quotes that OPINIONFINDER confidently labelled as subjective were of-

Topic	Label			Boundary		
	P	*R*	*F*	*P*	*R*	*F*
Abortion	35	61	44	34	79	47
Carbon tax	43	54	48	37	93	53
Immigration	17	52	26	19	49	28
Reconciliation	49	63	55	48	85	61
Republic	53	59	56	49	89	64
Same-sex marriage	43	60	50	40	72	51
WorkChoices	34	39	36	33	63	44
Average	39	55	46	37	76	50

Table 6.7: Results for the pilot experiments in detecting polar quotes using the context annotations. Label uses the labels from OPINIONFINDER and Boundary uses the learned confidence boundary.

Topic	Label			Boundary		
	P	*R*	*F*	*P*	*R*	*F*
Abortion	28	62	39	27	71	39
Carbon tax	25	59	35	22	68	33
Immigration	14	53	22	16	42	23
Reconciliation	28	64	47	36	76	49
Republic	44	60	50	42	79	55
Same-sex marriage	33	62	43	30	72	43
WorkChoices	20	39	26	20	54	29
Average	27	57	37	28	66	39

Table 6.8: Results for the pilot experiments in detecting polar quotes using the no context annotations. Label uses the labels from OPINIONFINDER and Boundary uses the learned confidence boundary.

ten considered to be neutral by our annotators. This tells us that subjectivity alone does not necessarily indicate that a quote will be relevant and opinionated. The converse is also true, as speakers will often cite facts that back up the argument they are trying to make. For example a proponent of action on climate change might cite a report on the potential dangers of a changing climate, while an opponent might discuss the economic cost of action. This would mean a largely objective quote could be considered evidence of an opinion.

The best average result when considering the context was 50%, while excluding the context reduced it to 39%. This indicates that the context the quote is placed in can affect its interpretation. This may be due to journalists asserting that someone holds an opinion, as in the following example:

> Ms Bryce, the Queen's Australian representative, will step down in March, and used a Boyer Lecture on Friday evening to speak in support of same sex marriage and a republic.
>
> She said she hoped the nation would evolve into a country where "people are free to love and marry whom they choose".

While it is fairly clear that the quote in the above example is polar, it is certainly made much clearer by the journalist's assertion that Ms Bryce is speaking in support of same sex marriage. Another common occurrence that changed the interpretation of quotes, was when speakers criticised other people in the debate. When the annotators considered this out-of-context it would not be clear which side the person being criticised supports, and thus who the speaker supports. When they were allowed to use the context, however, the interpretation of these quotes as polar became much clearer.

The topics with the highest F-scores were *Republic*, *Reconciliation*, and *Same sex marriage*. These topics also had the lowest proportion of neutral quotes. We speculate that this is largely because it is difficult to use facts to support a point of view in these debates, as the republic and reconciliation were largely symbolic movements, while same sex marriage is a question of morals. By contrast, *WorkChoices* and *Immigration* had the highest proportion of neutral quotes and the lowest F-scores. For WorkChoices, the debate almost required factual arguments, as it was about which system would produce better prosperity for the country, while immigration into Australia proved largely uncontroversial during the period covered by the documents, with the main point of contention being how many immigrants to accept per year.

6.4 Applications

Systems performing the task described in this chapter could be used by researchers to gain insight into how the reported opinions on a given topic change over time. News providers could use the system to collect a large number of quotes on a particular issue, with minimal curation, which could then be displayed or visualised on their website. In particular, they could identify instances where people have changed their opinion on an issue, which could be shown to users.

This corpus might also be useful in comparing quotes across different news providers. In particular, where two news organisations have well-known and opposing ideological leanings, it would be interesting to see if the quotes they use are largely the same, and if so, how the context they are placed in affects their interpretation. As a further experiment, it would be interesting to see how the reported quotes compare with the remainder of the source document or speech they are taken from.

We also believe that the corpus will be useful in furthering research into opinion mining. In particular, by labelling opinions at a slightly higher level than the MPQA corpus, we have built a corpus that could be useful in researching how fine-grained expression-level opinions, such as those produced by OpinionFinder, combine to form higher-level views on a topic. Some work towards this has been done by Johansson and Moschitti (2012), who used features derived from OpinionFinder output to perform document and aspect-level sentiment analysis over product and movie reviews. We believe that a similar approach could yield interesting results over our data set. In particular, the co-occurrence between our labels and the textually-anchored targets of opinion found using OpinionFinder, might provide insight into the aspects that are present for the topics that we cover.

6.5 Summary

In this chapter, we have examined the problem of annotating opinions in news articles. We proposed to exploit quotes, as they are used by journalists to provide evidence of opinions. We then proposed to label these quotes as *supporting*, *neutral* towards, or *opposing* a position statement, which is a statement of a particular point of view on a topic. This allows us to avoid the ambiguities that arise when considering a polarity label towards a given topic.

We next defined an annotation scheme and annotated a corpus, which enables us to test systems performing the task we have proposed. This corpus covers seven topics, with 100 documents per topic, and a total of 2,228 annotated quotes. As well as creating this corpus, we have identified many of the challenges that are inherent in creating an annotated corpus of opinions. We have further shown that the key challenge for annotators is identifying when a relevant opinion is being expressed.

Finally, we present pilot results on this data. The results show that we can achieve an average F-score of between 37% and 50% for detecting when a quote expresses a relevant opinion. The difference between precision and recall in this result indicates that to achieve better performance and to determine which side the speaker supports would require more information than just subjectivity.

7 · Conclusion

In this thesis, we have examined three core tasks: extracting quotes, attributing them to their speaker, and classifying the opinions they hold. Extracting speech may initially appear to be a simple task, however it turns out that there are challenges that prevent naïve solutions from working. The obvious approach of extracting content between quotation marks has surprisingly low precision, due to the presence of scare quotes, quoted titles, mixed quotes, and occasional instances of malformed text. Moreover, direct quotes, i.e. quotes surrounded by quotation marks, account for only 52.6% of all reported speech in our main corpus, the SMHC. Instead, this thesis shows that treating the task as a tagging problem, much like named entity tagging, produces better results than rule-based approaches, and allows us to extract a wider variety of reported speech.

For some applications it may be sufficient to simply extract speech. However, in most cases we would like to know who actually spoke the words in question. This task is known as quote attribution, and this thesis describes both a corpus and state-of-the-art approaches. Again, it may appear that this is a simple task, as many instances of speech are marked with a speech verb, such as said, exclaimed, wrote, and so on. However, this thesis shows that rule-based approaches are often thwarted. Even syntax-aware approaches that find the subject of the speech verb are insufficient. Our results show that supervised approaches with a rich feature set can vastly outperform these rule-based methods.

The final area that this thesis examines is opinion mining. While most work on opinion mining has classified sentiment at a document, sentence, or phrase level, we instead classify each instance of reported speech. This has several advantages. Most notably quotes are selected by journalists to provide evidence of a speaker's opinion, which means that the opinion should be reasonably clear. Annotating opinions is a difficult task, however our approach achieves moderate agreement, and so we have built a corpus using this approach. While this thesis does not present approaches to the full task, we do present preliminary results in identifying quotes that provide evidence of an opinion. In the following sections we will review the contributions of this thesis on each of these tasks individually.

7.1 Quote Extraction

In this thesis, we have presented the first large-scale experiments on the full task of quote extraction. As part of this we have introduced a fully-labelled corpus of news articles (SMHC) with annotations of direct, indirect, and mixed quotes. We have also evaluated our methods over another corpus (PARC), which is not yet fully labelled. The SMHC and these evaluations are major contributions of this thesis, as previous work either evaluated quote extraction jointly with quote attribution, or did not consider indirect and mixed quotes.

We have also presented two approaches to quote extraction, which we compare against two baseline methods. The first approach treats the task as a tagging problem, where we train a CRF model over the sequence of words with IOB class labels. The second approach is based on the constituent parse of each sentence, where each constituent is labelled according to whether its span represents a quote or not. A binary maximum entropy classifier can then be trained over these nodes. These two methods are compared to a simple lexical baseline and a baseline that returns the object of every speech verb.

Our results show that the token-based CRF model outperforms the constituent-based maximum entropy classifier and the baseline methods. The token-based method achieves an F-score of 78% in terms of exactly matching the span of quotes, and a score of 89% for a metric that rewards partially correct answers. These methods and results are state-of-the-art, and should provide a solid foundation for future research. There are also many practical applications that can benefit immediately from this research.

7.2 Quote Attribution

In addition to our experiments on quote extraction, we have presented a corpus, methods, and extensive evaluations on quote attribution. Quote attribution is the task of matching quotes with entities in the text who represent the quote's speaker. The corpus we built covers the same 965 news articles as the quote extraction corpus, so includes attribution for all direct, indirect, and mixed quotes, which is a novel contribution of this work. At the time of writing this is the largest fully-annotated corpus of quotes.

We have also presented and evaluated several methods for this task. The methods include an extension to the work of Elson and McKeown (2010), where we removed some of their features that rely on gold-standard information. Instead, we proposed to use sequence tagging methods, so that we could generate realisable versions of these features. We also introduced a range of novel features for this task that improve the results substantially. We tested these features with one existing and one novel class-labelling scheme, with the novel scheme permitting the training of a full CRF sequence model. Our methods are evaluated over four corpora, with two from the news genre and two from literature. This is a key contribution of this work, as previous work has mostly included small-scale evaluations over a single corpus. Crucially, we provide a detailed comparison with previous work.

While these evaluations provide a solid basis for comparisons with future research, they all use the set of candidate speakers that is provided with the corpus. This is problematic as the different corpora make different assumptions about what it means to be the speaker of a quote. Furthermore, if speaker annotations are linked to the provided set of mentions then there is a challenge in evaluating quote attribution systems that use a different set of mentions. In Chapter 4, we discuss this problem in detail and present two alignment methods that allow us to perform these evaluations.

Next, we present an evaluation of the full pipeline of mention detection, coreference resolution, quote extraction, and quote attribution. This evaluation covers the full set of quotes, and tells us how well we would expect the full system to do in realistic conditions. Finally, we present a further evaluation that looks at quotes whose speakers can be disambiguated back to a KB entry by a NEL system. This full task is challenging, as it sits at the end of a long pipeline of NLP systems. As such, the results are fairly modest, however this does present an interesting area of future research.

Another contribution from our work in quote attribution is that we show how quote attribution can be used as an extrinsic evaluation of coreference resolution. We achieve this by evaluating to what extent each coreference resolution system benefits quote attribution, using the two alignment methods. Though the alignment methods are imperfect and allow certain edge-case systems to cheat the task, they still give some insight into the efficacy of coreference resolution. In particular they allow us to examine how well coreference resolution systems perform in literature, which has not been previously studied due to a lack of annotated data. The results of this evaluation show that the coreference resolution systems we evaluated are not effective on literature, with mixed honorifics being a particular area of concern.

7.3 Opinions

In our opinion mining work we focus on quotes, as they are typically used in news articles to provide evidence of an opinion. The trouble with quotes, and with news articles more broadly, is that they are on diverse topics, where simple polarity labels, such as *positive* and *negative* do not necessarily make sense. In place of these labels, we propose to use *position statements*, which clearly state a point of view on a given topic. Quotes can then be labelled as *supporting*, *neutral*, or *opposing* a position statement.

Using this formulation of the task, we create a novel opinion corpus which covers seven topics, with 100 documents per topic, and a total of 2,228 annotated quotes. As well as building this corpus, we have identified and described many of the challenges that inhibit annotator agreement on this task. We have further shown quantitatively that the key challenge for annotators is identifying when a relevant opinion is being expressed, rather than identifying which side an opinion is on. We conclude our work on opinion mining with some preliminary results in identifying when a quote shows evidence of an opinion.

7.4 Future Work

This thesis examines three main tasks, so in this section we will discuss the future work for each of them individually, before discussing the work that would be required to more fully achieve the overall goal of this work.

7.4.1 Quote Extraction

For quote extraction, this work revealed several areas of future research. First and foremost, Chapters 2 and 4 identify several differences in performance between SMHC and PARC. We speculated that these differences were largely the result of the incomplete data in PARC. Testing this idea would simply involve running the same experiments over the fully annotated version of PARC. If the differences persisted we would need to find some other explanation.

It would also be worth performing a more thorough analysis of the properties and function of quotes, and how they differ between corpora. In particular, in Chapter 2 we compared the distribution of quotes in each corpora, however the conclusions we could draw from this were limited, as PARC is not yet fully annotated. Once PARC is fully annotated a more comprehensive comparison will be possible.

In terms of the methods, we believe that it would be advantageous to use a parser more directly for this task, as we suggested in Chapter 2. This would involve training a parser on data with extra annotations that indicate where quote boundaries are. We expect that this type of approach would outperform both the constituent and token-based approaches that we have outlined here.

7.4.2 Quote Attribution

Our work also raises some interesting questions about quote attribution. Our experiments in Chapter 3 showed that the choice of model made little difference, however Chapter 4 showed that the choice of candidate speakers can have a fairly strong impact. As such, we believe that a dedicated candidate selection process could yield further improvements in performance.

The interaction between quote attribution and coreference resolution is also an interesting area of future research. Our work in Chapter 4 showed that canonical mentions of an entity are not always clustered with their later references. It may be that a joint quote attribution and coreference resolution model could help to cluster these mentions correctly, while potentially improving quote attribution in the process.

In addition to examining coreference resolution for the speaker of each quote, it would be interesting to look at coreference for the mentions that occur within quotes. In the two literature corpora these mentions are often going to be important in understanding dialogues, while in the news corpora the mentions will be important in identifying the topic of the quote. Correctly resolving these mentions would likely form an important part of any system performing the opinion mining task that we describe in this work.

7.4.3 Opinion Mining

While this work has presented complete systems that can find quotes and attribute them to their speakers, the work on opinion mining is still preliminary. We have presented early experiments in detecting polar versus non-polar quotes, but this work does not include experiments that complete the main opinion task that we described, namely in identifying support, neutrality, or opposition to a position statement. In this section we will describe the work needed to fully complete the goal of this work.

The first challenge that needs to be overcome is the identification of position statements. As noted in Chapter 6, we believe that the overall task should be semi-supervised, so position statements would need to be manually created for each topic of interest. This also means that a small amount of data should be manually annotated for each position statement. One potential short cut for this requirement would be to use data from online debate sites.

Once a position statement and data have been created or acquired, we would need to identify documents with relevant quotes. In this work we used Solr with manually-created search queries. This produced reasonable results with little effort, so we view this as a viable approach. Future research could investigate automatically finding documents based on position statements, but we expect that this would be a lot of work for a relatively small gain.

The next stage in the process would be to run NEL, quote extraction, and quote attribution. The output of these systems gives us the quotes and their speakers, so we next need to ensure that the individual quotes are valid and on-topic. While completing our opinion annotations we found many quotes that were invalid, either because they were titles or because they were attributed to an organisation. Our evaluations in Chapters 2 and 4 show that our improved quote extraction system will dramatically reduce the number of invalid quotes, so a dedicated system here is unnecessary.

Determining if quotes are on-topic, however, remains an open problem. Our preliminary experiments show a possible direction for future work, but do not achieve performance that would be acceptable in practise. Further investigation into the properties of polar versus non-polar quotes is necessary.

The final problem that needs to be solved is identifying whether quotes support or oppose the given position statement. We expect that progress could be made by combining the output of general opinion mining systems with problem and position statement specific information. In particular, we stress that it is our expectation that a small amount of labelled data would be necessary for each position statement. Investigation into using data from online debate forums may also yield progress on this problem.

7.5 Summary

Our work examines the problem of automatically detecting and attributing quoted opinions in news. As part of this work we have established clear baselines, data, and methods for both quote extraction and quote attribution. We have also built a corpus of quotes that are labelled according to which side of a debate they support. This work has established accurate systems for these tasks, which can be used for immediate applications.

A · Quote Extraction and Attribution Scheme

This appendix is the quote extraction and attribution annotation scheme that was used by annotators in the creation of the Sydney Morning Herald Corpus.

A.1 Introduction

We are interested in building an automated system that can find out who said what in news articles. The system will be able to find direct, mixed, and indirect quotes, and will then attribute those quotes to named entities that are found in the article. In order to do this we need several hundred example documents that can be used to both inform the system and to evaluate how well it performs. This guide will clarify exactly what we need annotated, and will include instructions on using our annotation system. Throughout this guide we will use *italics* to indicate the content of a quote and **bold** to indicate the source of the quote.

A.2 Overview

There are three types of quotes that we are interested in finding. They are:

1. Direct quotes, which are literal transcriptions of what was said, marked with quotation marks

2. Indirect quotes, which are paraphrased forms of what was said, with no orthographic marking

3. Mixed quotes, which are part indirect and part direct

As part of this we also need to identify who said the quote, which we will do in two different ways. First we are interested in the **source**, which is the span of text that indicates who the speaker of a given quote is. Consider the following two examples where the source is emphasized in **bold**:

> *"It's unusual that they'd ask for a closed venue and then turn up at our training,"* McKinna said.
>
> *"I'm very sorry for the situation that has been created,"* he said.

For both of these cases it is useful for our system to know that there is a span of text that refers to the speaker. However these annotations do not actually tell us who the speaker is. For the first example there could be multiple people with the name McKinna, and for the second example the use of a pronoun makes the speaker's identity ambiguous. Furthermore some quotes will not include an explicit source in the text.

We need to be able to resolve the speaker of each quote back to a named form that is found somewhere in the document. Since speakers will frequently have many quotes in an article, we will first group quotes into *chains*, where each chain will include all of the quotes by a given speaker. For each chain you will be given a list of all the named people in the article, which you can use to choose the correct speaker. Gathering both these pieces of information will let us know who the speaker of each quote is, as well as how each quote is syntactically attributed to its speaker.

A.3 When to annotate

This section will outline when you should decide to annotate a quote, while exactly what you should annotate will be covered in later sections.

A.3.1 Direct quotes

A direct quote is a literal transcription of what a person said, which is enclosed in quotation marks, i.e. the " and " symbols. While these quotes will in general be easy to identify, it is worth noting that not everything enclosed in quotation marks counts as a direct quote. For instance, quotation marks can be used as "scare quotes", to introduce some terminology or to distance the author from some phrasing, for example:

> The greyhound won his Healesville heat in the second fastest time of the day and is a live chance in the "straight track" 340 metres scamper.

Here "straight track" is scare quoted, and not a direct quotation, so you should not annotate examples like this. If you are unsure it can be helpful to look for the presence of a speech verb, such as said, reported, replied, and so on. If you can't find one then there's a good chance that what you are considering is not a direct quote.

A.3.2 Indirect quotes

Another way to introduce speech is to use an indirect quote. Indirect quotes are paraphrased versions of what a person said, and include no orthographic markings to indicate their extent. As such they will be much more challenging to annotate, and so we will need to define much more clearly what we are looking for. We will start by looking at an easy example:

> He said *that the policy would cost thousands of jobs.*

The key things to note about this example are:

1. The writer is reporting on some content that was communicated, in this case indicated by the verb said
2. The content of what was said is present in the text

These can be considered your criteria for annotating a quote, and you should not annotate quotes that do not meet them. Consider the following examples:

> He believes that the policy would cost thousands of jobs.
>
> He talked about the policy's impact on jobs.

Despite the content being the same, we don't want you to annotate cases like the first example as there is no communication event being reported on. Instead, the writer is reporting on a person's belief, without explicitly indicating that the person expressed it to anybody. In the second example the writer is indicating that speech took place, and even includes the topic (i.e. the policy's impact on jobs), but does not directly include the content of what was said. You should not annotate examples like the two above. Note that the distinction between text that includes the content of what was said versus text that includes a description of the content will often be difficult to to make.

A.3.3 Mixed quotes

Some quotes will include portions that are directly quoted, and portions that are indirectly quoted, as the following example shows:

> He said *that action should be taken "to reduce the national debt and restore the confidence of banks."*

In this case the journalist has paraphrased an early part of the quote (i.e. that action should be taken), while still including a directly quoted portion at the end. In these instances you should annotate both the direct and indirect portions as a single quote. Often, only very small phrases will be directly quoted, such as:

> He said *that the national debt had become "frankly enormous" and that action should be taken to reduce it.*

In another context, "frankly enormous" would be understood as a scare quote (not part of a quote), however, because of the way it's attributed in this sentence, the reader is to understand that the speaker did use that specific phrase, so it should be included as part of a mixed quote.

A.4 Content

Once you have decided to annotate a quote you need to decide exactly what you will be marking. For direct quotes the span that you should mark will be made obvious by the quotation marks. You should include everything between the quotation marks as well as the quotation marks themselves. Indirect and mixed quotes present more of a challenge. Broadly speaking you should only annotate text that is part of what the person actually said, and avoid annotating things that the person only may have said. In this section we will clarify some of the subtler parts of the annotation.

A.4.1 Inclusion of *that*

Let's look again at the example from above:

> He said *that the policy would cost thousands of jobs.*

The part that the speaker actually said is: *the policy would cost thousands of jobs*. However, note that we have included the word *that* in the example. The reason for this is that the word *that* plays an important role (which we will see in the next subsection), and so you should include it when it is present, even though it is not part of what the speaker said. Note that in the case of direct quotes you should not include *that*, unless it is contained within the quotation marks.

A.4.2 Coordination

Including *that* becomes particularly important when we have multiple statements joined with an "and" (or similar coordinating word). Consider the following two examples:

> He said *that Lord's would bid aggressively to host a neutral Test between Pakistan and Australia next year*, and he wanted the ICC to reconsider a global Test championship, perhaps in two tiers, to ensure matches always have some relevance.
>
> He said *that Lord's would bid aggressively to host a neutral Test between Pakistan and Australia next year, and that he wanted the ICC to reconsider a global Test championship*, perhaps in two tiers, to ensure matches always have some relevance.

The inclusion of that in the second example makes it clearer that the quote covers a wider span, rather than just the part before the and. This should serve to strengthen the evidence that what the person said continues on to the next statement, however it should not be considered definitive. In some instances the word that will not appear, but it will be clear that additional text is still part of the quote's content..

A.4.3 Fragmented quotes

In some cases quotes will be fragmented into several parts by an intervening clause, as in the following example:

> *"Ownership of Australian businesses by state-owned enterprises is an inherently unhealthy thing,"* Senator Joyce said, *"it raises all sorts of complications."*

In this case simply annotate the two content spans as separate quotes. Note that you will need to specify the source separately for each of the quotes.

A.4.4 Punctuation

Most punctuation that is adjacent to the content should be included within the span. For direct and mixed quotes you should always include the quotation marks as part of the content. Sentence-ending punctuation, such as full stops, questions marks, and so on, should be included if the quote ends the sentence, which will almost always be the case. Commas adjoining the content should *not* be included (unless they are enclosed within quotation marks), as they generally do not play a role within the quote, rather they are playing a role in the sentence the quote is found in. Other punctuation should be included or excluded based on whether they are considered part of the content of the quote.

A.5 Source

In addition to the actual content of quotes, we would like you to annotate the source of the quote. The source is a span of text that is syntactically related to the quote, that indicates who the quote is attributed to. In most cases it will be a noun phrase that refers to a particular speaker, as in the following examples (source marked in **bold**):

> *"Ownership of Australian businesses by state-owned enterprises is an inherently unhealthy thing,"* Barnaby Joyce said.

In addition to the proper noun example above, sources will frequently contain pronouns or common nouns as in the following examples:

> *"Ownership of Australian businesses by state-owned enterprises is an inherently unhealthy thing,"* the senator said.

> *"Ownership of Australian businesses by state-owned enterprises is an inherently unhealthy thing,"* he said.

When annotating a nominal source we would like you to annotate the entire noun phrase, including determiners (the, a, an, etc.), adjectives, positions, titles and so on. We do not, however, want you to annotate appositions or relative clauses, which will usually be split from the main noun phrase by a comma.

There are a number of other considerations that you should be aware of when annotating sources:

- The source might also include inanimate objects, such as reports, letters, submissions and so on.
- A quote may have no source.
- In some rare cases there will be a non-nominal source.
- For fragmented quotes you will need to annotate the source separately for each fragment. It is important that you do this, as it helps us to join these fragments up later on.
- The source might overlap some of the content of a quote.

A.6 Quote chains

When you are annotating a quote, you will need to include it in a chain, which is a set of quotes that are all by the same speaker. All of the quotes by a given speaker should be included in the same chain, regardless of how they are syntactically attributed to the speaker. In other words you should be adding a given quote to a chain based on who you understand said the quote, not based on the specifics of the source. You can create a new chain if there is no chain for the speaker of the quote you are annotating.

A.6.1 Speaker

The speaker is the person who actually said the quote, who should be identified from your reading of the document. It might not be mentioned explicitly where the quote is presented, but it should almost always be clear from the context. You'll be given a list of suggestions to choose from, as other annotators have marked all the named people, locations and organisations in the document. There are also two additional options, one for when there is no speaker and another for when the speaker is missing from the list.

Most speakers will be easy to annotate, however there are some edge cases:

- If the speaker is not explicitly named in the document they will not appear in the list, so in these cases choose **missing** as the speaker.

- In some cases there will be spokespeople, both named and unnamed, who are saying something on behalf of another person or organisation. In these cases the speaker is always the spokesperson, even if they are unnamed.
- Occasionally there will be multiple options for an entity in the list (this is a result of annotators making mistakes in an earlier process). When this happens choose the option that is the most specific about the entity, and the first option if there are multiple options that are equally specific.

A.7 Difficulties

There are several other issues that will occasionally crop up, which you will need to know how to deal with.

A.7.1 Hypothetical quotes

Sometimes journalists will suggest that somebody might say something, but we don't actually know if they did say it or not. For example:

> He may well say *that he regrets his decision.*

In the above example the person did not necessarily say anything, but the journalist has proposed that the person might have said something. For these cases please annotate the quote as you would any other quote. This includes marking the source and speaker as the person who might have said something.

A.7.2 Nested quotes

Sometimes quotes will include quotes within themselves, such as in the following example:

> *"They're just doing it well each week and they're basically saying, 'We're going to play like this, if you're going to beat us, you've got to match us' - and nobody can match them at the moment."*

For these cases please annotate only the outermost quote, i.e. the quote with the largest content span.

A.7.3 Written documents

Excerpts from written documents should have their content annotated similarly to other quotes. However attributing these quotes can be more difficult. The source should generally be indicated syntactically as in the following example:

> *The investigation and Australian operational records indicated "there was no substance" to the claim*, the defence report said.

The speaker for written documents should be the most specific reference to the organisation it originates from, and should be marked as missing if that reference is not in the list. This will become important when there are committees within organisations that have authored reports. For these cases you will need to mark the committee as being the speaker, or missing if the committee is not in the list. In some cases the author or authors of a document will be known. You should mark any quotes from the authors as separate chains to the quotes from the report or document.

A.7.4 Organisation versus spokesperson

In some instances you will have quotes from an organisation closely followed by quotes from a spokesperson or a member of the organisation, as in the following example:

> In the past fortnight, the bureau said *many temperature records had not so much been broken as smashed*, as the southern states endured the heatwave. *"Normally you see records broken by a fraction, but in Tasmania for example, the record at one station went by nearly 2 degrees,"* said Dr David Jones, head of the bureau's National Climate Centre.

In this example and others like it you should mark the two quotes as being in separate chains, despite the fact that Dr David Jones is speaking on behalf of the bureau.

A.8 Tool notes

The annotation tool consists of a document window, which contains the text of the document, and a quote chain panel on the right. The document is given to you with all text within quotation marks already annotated, although you will still need to annotate the source for each quote. Some of the direct quotes that have been annotated will actually be mixed quotes, so you will need to delete those spans and recreate them. Occasionally the existing annotations will be incorrect and should be fixed when found.

You will find that the text of the document is broken up strangely, with extra spaces inserted in some words and around punctuation. This is a standard form of linguistic preprocessing that you should ignore. You will also notice that the tool will force your selection to be over whole tokens, rather than allowing parts of words to be annotated. This is by design, as parts of words usually don't make sense as part of a quote.

A.8.1 Annotating a content span

To annotate a content span simply highlight the text that you want to annotate using your cursor as you normally would. You can then right click on the span and select either "New" to create a new quote chain, or a button coloured with the colour of an existing quote chain. The text of each coloured button will be part of the content of the first-annotated quote in a chain.

A.8.2 Annotating a source

To annotate a source, first select (i.e. click on) the quote that you want to add the source to. When you do this the quote will become highlighted. You can then highlight the span of text that represents the source as you normally would. You can then right click and select "Add Source", which will add the highlighted span as the source of the given quote. The source span will be highlighted similarly to the content span when the quote is selected, so you can verify that you've annotated the correct span by clicking on the quote.

A.8.3 Selecting a speaker

Lastly you can choose a speaker for each quote chain in the menu on the right. The options in the menu will be highlighted in their chain's colour, and will include a short fragment from the first quote you annotated in that chain. Under each of these options there will be a drop down menu, which allows you to choose the speaker. The direct quotes that were already annotated will have their speakers selected already, but you will need to choose a speaker for all of the new quote chains you created, and you may need to fix errors in the existing annotations.

A.9 Further examples

In this final section of the guide we will include some difficult examples, along with their correct annotations. We can't give examples for every difficult case, but these should cover a few of the more common cases that you might encounter.

> He said *that the national debt had become "frankly enormous" and if Britain failed to retain the confidence of banks, higher interest rates would stymie recovery.*

The end of this quote is difficult to determine because of the "and" in the middle, however it appears that the journalist intends us to understand that the quote continues and includes the rest of the sentence.

> Instead, *the committee will inquire into "the international experience of sovereign wealth funds and state-owned companies", and will also investigate "their role in acquisitions of significant shareholdings of corporations".*

This example contains no cue to indicate that there is a quote, aside from the presence of quotation marks. As such the extent of the content is very difficult to determine, and the source is highly ambiguous. When this is the case it is better to avoid marking the source, as the only candidate is the committee, and we can not be sure that the committee is actually the speaker. For the content we have marked most of the sentence, as marking less would result in a span that does not make sense on its own.

> Captain Chris Gayle insisted *that he was up for the challenge*, saying: *"No captain wants to lose, so you have to be a strong individual in these circumstances."*

In this construction there is an indirect quote, followed by a speech verb, and then a direct quote. This type of construction is quite common and will often be ambiguous. The direct quote will be easy to annotate, although it is worth noting that in this case you should still mark Captain Chris Gayle as the source of the direct quote. The indirect quote will depend very much on the content and the verb used. In this example we have annotated the indirect part, as the verb insisted indicates speech, and the content is plausibly something he said. In other instances you will find that the first part is not indirect speech so much as a broad description of a topic, while the direct quote will be the actual content. For those cases you should not mark the indirect quote.

> Mr Hatoyama spoke to a more obviously spontaneous assembly, arguing for change: *"What the American people have done, the Japanese people can do too".*

This example includes a span of text that should not be marked as it is a description of the topic under discussion rather than indirect speech.

> *"But considering over a quarter of Australians think that breastfeeding in public is unacceptable, we know there is a long way to go,"* Dr James, a course co-ordinator in the Department of Nursing and Midwifery at RMIT University, said.

There is apposition in this example, which is the text a course co-ordinator in the Department of Nursing and Midwifery at RMIT University. This apposition should not be included as part of the source.

> *"Based on our experience at this point in time, we see a greater risk in the immediate term of a serious funding gap arising in the context of CMBS, rather than from foreign bank activity,"* ANZ's submission said.

Here the source includes the possessive ANZ's, which is included in the source. Furthermore the source is inanimate as it is a submission. The speaker for this example should be ANZ if it appears.

> She welcomed the recommendation for more support for mothers to begin breastfeeding. *Too often babies were taken away for jabs and checks after delivery, at odds with newborn babies' "extraordinary capacity to find the breast, attach and feed".*

In the first sentence the pronoun She appears, and is clearly referring to the speaker of the second sentence. However it should not be marked as it is in a different sentence, and thus not in a syntactic relationship with the content. The second sentence contains no cue aside from quotes, but it is implied to be something that She said, so it should be marked as a quote.

> *"Ownership of Australian businesses by state-owned enterprises is an inherently unhealthy thing,"* Senator Joyce said. *"It raises all sorts of complications."*

This quote is fragmented. Both fragments should be annotated separately, although only the first one should include Senator Joyce as a source.

> AFIC posted a letter on its website yesterday. *"The government spent a great deal of time, effort and money to persuade many retail investors to participate in the three Telstra share offers,"* the letter reads. *"At the time I acquired my shares in Telstra there was no prospect of the government taking such unprecedented steps to attach the company and its shareholders."*

This quote is fragmented and contains an inanimate source. Both content spans should be annotated, although only the first one should have the source the letter. The speaker for the two quotes should be AFIC.

A.10 Need more help?

This guide is intended to get you started with the annotation project, but it is impractical to have it cover all of the weird and wonderful constructions that you will find when annotating. If you need help with either the annotations or the tool then don't hesitate to contact us!

Acknowledgements

This guide was based on an earlier guide on annotating direct quotes written by Matthew Honnibal. It was also designed with reference to Silvia Pareti's guide, so as to ensure comparability.

B · Opinion Annotation Scheme

This appendix is the opinion annotation scheme that was used by annotators in the creation of the opinion corpus.

B.1 Overview

The purpose of this guide is to help you make the most consistent annotations possible. There are many factors and ambiguities that could lead to confusion and disagreement amongst annotators about the best choices to make, including:

- How much the article's content influences your decision about a quote
- The speaker's previous quotes, both in and out of the article
- Your own nationality, background, and opinions on the topic the quote is related to
- Your background knowledge about the speaker of the quote, either outside of this task or from previously annotated articles
- The style of language of the speaker being difficult to interpret (particularly true of politicians)

This guide aims to clarify how you should resolve the above ambiguities and how you should use the tool more broadly. For any cases that this guide does not clarify, please contact us and we can help you out.

In some cases an entire document will be off-topic. If this is the case please check the checkbox labelled "Article topic incorrect", which appears on the right-hand side of the screen, and then the "Save" button to move to the next document.

B.2 Quick Reference

Annotating a quote involves the following steps:

1. Check that the quote is on topic, and that it is actually a quote
2. Check whether the speaker is correct
3. Read the full quote, including the indirect portions
4. Is this quote evidence for the real speaker's position on the topic, assuming no knowledge of the topic?
5. What if you use your full knowledge about the topic?
6. Double check that you have selected everything you intended to and if so, move on

B.3 Annotation Process

B.3.1 Check that the quote is on topic, and that it is actually a quote

There will be some quotes where you can't choose a sensible sentiment score, usually due to one of the following reasons:

The quote in question was not spoken by a human speaker. A good example of this is a quote from legislation or some other document (surveys, newspapers) with no author attributed. In these cases there is no human speaker to hold a particular opinion or sentiment, so you should mark these cases with the checkbox "it is not a quote".

The quote is unrelated to the topic of the article. For some cases the quote may be completely unrelated to the topic we are interested in. For example, the quote "I had a sandwich for lunch" is very unlikely to have anything to do with the question of whether Australia should become a republic. You will find some more borderline cases in certain types of articles, however we expect these cases to be fairly rare, and we would like you to use this option sparingly. To mark a quote as being off topic check the "Is not related to the topic" checkbox.

You may find some other strange cases that don't fit into any of these categories. We have included an ability to add comments to mark any cases where you're completely unsure of what to do. You should notify us when you have left comments, so that we can go back and sort out the problem. Note that you should only use the comments section to mark problem cases, not to justify your decisions.

B.3.2 Check whether the speaker is correct

Sometimes the quote attribution system will not detect the speaker of the quote (it will be listed as "null") or the quote will be attributed to the wrong person. In these cases please tick the box indicating that the quote has "an incorrect speaker", and then annotate the quote as though our system had got it right. As this is a new and experimental system we expect it to be wrong about the speaker about 30%-40% of the time, which is why we still need the sentiment marked in these cases.

B.3.3 Read the full quote, including the indirect portions

Before you can annotate the sentiment you need to be clear on what constitutes a quote in this research. Our system will highlight any text that appears between quotation marks, provided there are at least four words in the quote. In the simplest case each highlighted span of text is a quote, however, it is quite common for journalists to paraphrase part of what a person said, in order to make it briefer or simpler. At the moment our system is not able to pick this up, **but you should still consider it as part of the quote**, provided it is in the same sentence. Consider the following example:

> Just a short distance away from the Australians for Constitutional Monarchy lunch, the Prime Minister was insisting that a republic was only "a matter of time."

In the above example, the directly quoted text, *"a matter of time,"* doesn't really make sense on its own. However it is clear that the Prime Minister said something like *"the republic is only a matter of time."* The part that does not appear between quotation marks is called an indirect quote and you should include this indirect part as part of the quote when you annotate it for sentiment, **even though it wont be highlighted**. In some cases doing this correctly will be very important, as shown in the examples below.

Example 1

> The chairman of the Australian Republican Movement (ARM), Mr Malcolm Turnbull, said that while it was too early to claim victory, "I think clearly the republicans will have a majority at the convention"

In this case the indirect part of the quote that should be included is: while it was too early to claim victory. You **should not** include the text The chairman of the Australian Republican Movement (ARM), Mr Malcolm Turnbull, said that, as the journalist is not indicating that Malcolm Turnbull said any of that. In particular the fact that he is the chairman of the ARM is not something that the journalist is indicating Malcolm Turnbull said, and so you should not consider that to be part of the quote.

Example 2

> Sir John Gorton said he was "very underwhelmed" by the referendum, although he did not see that "Australia has to stay with the Queen of England forever"

This quote demonstrates the importance of including the indirect parts of the quote. The directly quoted part sounds as though the speaker strongly supports the monarchy, but when the indirect portion is added, the quote becomes much more neutral. So, to clarify, the text that should be considered part of the quote is: while it was too early to claim victory, "I think clearly the republicans will have a majority at the convention"

Example 3

> Mr Photios said Princess Diana represented the future of the British and other monarchies. With her sons, she defined how the monarchy could adapt to reflect progressive values and was "the embodiment of the transition that is happening to our constitutional monarchy"

This example shows a case where the quote is spread over two sentences. In this instance you should only consider the indirect part from the sentence the directly quoted part appears in. The remainder should not be considered part of the quote. So, again to clarify, the part of the quote you can use is: With her sons, she defined how the monarchy could adapt to reflect progressive values and was "the embodiment of the transition that is happening to our constitutional monarchy"

Finally, there will be some cases where the journalist indicates that somebody said something, but they wont use any directly quoted text. You can ignore these cases as it is currently not a focus of this research.

B.3.4 Is this quote evidence for the real speaker's position on the topic, assuming no knowledge of the topic?

In order to help achieve consistent results, we will need you to mark the sentiment in two different ways. Please be aware that you need make a choice for **both** of these tasks for every quote.

For this part our goal is to find out how strongly a quote indicates a position on a topic, when little or no background knowledge or context is used. Suppose you are marking a quote by Tony Abbott on the topic of whether Australia should become a republic. The way to mark it is to pretend that you are trying to convince a friend of Tony Abbott's position in the debate. Assume your friend knows nothing about the topic or the people involved, and that you can only show them this single qoute (both the direct and indirect portions). Would the quote convince them of Tony Abbott's position? If so then you should mark it as such. You can perform this annotation using the top row of circular buttons.

Example 1

> In the Telegraph, Brendan Shanahan criticised the "snide commentary" against the Queen before declaring that Australia "will, and should, one day be a republic"

This should be marked as "Strongly or clearly supporting" the republic, because the speaker, Brendan Shanahan has declared that Australia should be a republic.

Example 2

> "everything that Hewson, Howard and Bishop are saying is really an insult to the intelligence of their audience"

Even though this quote expresses a strong opinion, it should be marked as neutral for this first task, as you must assume that you do not know who Hewson, Howard and Bishop are. Or, thinking about it another way, you cannot use this quote to convince someone of the speaker's position, as the person you are trying to convince may not know who Hewson, Howard and Bishop are.

B.3.5 What if you use your full knowledge about the topic?

For Part 5 of the annotation, you should annotate in a similar way to Part 4, but this time assume that your friend is knowledgeable about the topic and the people involved. This will influence your decision typically by allowing you to incorporate political affiliation, memberships to organisations and societies, previous statements and quotes, amongst other information. You can perform this annotation using the bottom row of circular buttons. You will find that most often you choose the same option here as you did for Part 4, however some choices will end up being different, so we'll show some examples below.

Example 1

> "everything that Hewson, Howard and Bishop are saying is really an insult to the intelligence of their audience"

This example is the same as from Part 4, however now you can annotate it as having an opinion, as it does provide evidence of the speaker's position, if you know who Hewson, Howard, and Bishop are.

Example 2

> Kate Mannix , told The Age it was a "shadowy group [whose] strategies are lifted from conservative US Catholic groups"

In this example (taken from the abortion task) you would have to mark this quote as "neutral" for Part 4, but for Part 5 you could infer that Kate Mannix is pro-choice, as she is criticising a conservative group (though the evidence here is certainly not strong or clear).

B.3.6 Double check that you have selected everything you intended to and if so, move on

Marking opinions and sentiment is a surprisingly difficult task, and our research requires high-quality data in order to produce good results. As such we ask that you take care that you have made the correct decisions, and in particular that you have marked what you intended to.

B.4 More details/FAQ

B.4.1 How do I interpret the markers on the sentiment scale?

The easiest way to describe which sentiment score to choose is by example:

Strongly or clearly opposes should be chosen when there is a very clear statement of opposition, or a very strong opinion that could be best interpreted as opposition to the position statement. In the "republic" task examples of these quotes could include statements like "republicans are morons", "The monarchy is the best thing to ever happen to Australia", "The monarchy is a well established system that will last forever".

Opposes should be chosen when there is evidence of opposition to the position statement, but it is not strong and not clear. From the "republic" task an example would be "I haven't really considered the issue carefully, but I think the monarchy is fine".

Neither opposing nor supporting should be chosen when there is no clear sentiment. This should be applied for statements of fact about the issue, for example in the "republic" task "Changing from a monarchy would require the support of all the states and the drafting of a new constitution". It should also be applied to clearly neutral statements like "I do not care if we are a republic or a monarchy", or statements that indicate positive or negative sentiment to both sides such as "I like the monarchy but I'd be fine with a republic too".

Supports should be chosen when there is evidence of support for a position statement but it is not clear and not strong. An example of this from the "republic" task would be a quote saying "I don't like the royal family, they don't represent this country well", or "The monarchy is a bit old hat".

Strongly or clearly supports should be chosen when there is a very clear, or a very strong opinion that could best be interpreted as support of the position statement. Examples of this include, for the "republic" task: "It is time for a change to a republic system. Australia is being held back by the monarchy", or "The monarchy is a dated, tired system which should have been left behind at Federation".

B.4.2 Help! This quote is really tricky and I can't decide!

For both Parts 4 and 5 there will be many quotes where making a decision is tough. In order to make these decisions easier we would like you to prefer the less opinionated option when you are faced with a tough choice. If you are unsure whether to annotate "neutral" versus "supports", choose "neutral", and also choose "neutral" when considering "neutral" versus "opposes". You should also prefer to annotate "supports" or "opposes" in preference to "strongly supports" or "strongly opposes". Attacks on people holding opposing views, such as "all monarchists are idiots", implies support for the other side, and should be annotated as such, even though there is no explicit statement of support for the opposite position.

B.4.3 What do I do when I've finished annotating a document?

On the right side of the screen you should see a button labelled "Save", which you will need to press to save your annotations. If you close or navigate away from the page before clicking the "Save" button, you will lose your annotations. You can save at any time and come back to a document later, though in general you will find it easier to do a document all at once. When you click "Save" and you have finished a document you should be taken to the next document.

B.4.4 What pitfalls are there?

You need to take care that you are not inserting opinions that are not there, based on your own views, or knowledge of the article and the speaker. For example after reading many quotes by Tony Abbot in the republic task, you may be inclined to annotate any quotes of his as being pro-monarchy. Alternatively, you might be a strong supporter of the pro-choice side of the abortion debate, and this might tempt you to mark more quotes as supporting your side. Both of these situations should be avoided. Remember to follow the process above and consider how effective a piece of evidence each quote would be in an argument over the speaker's position.

B.5 Topic-specific annotation guide

This section includes a summary of the topics covered in the annotation task. You can read the descriptions of the topics in the following section (you can ignore the others). These are intended to get you familiar with the basic topic under debate.

B.5.1 Abortion

Position statement: Women should have the right to choose an abortion.

Debates around whether abortion should be legal have been going on for decades, with social conservatives and religious groups generally being against abortion, while social progressives and libertarians have been in favour. In this debate those in favour of legalised abortion style themselves "pro-choice", while those against style themselves "pro-life".

B.5.2 Same sex marriage

Position statement: Same-sex couples should have the right to attain the legal state of marriage as it is for heterosexual couples.

The articles on this topic are not exclusive to Australia, since there is a strong relationship between Australia and the United States in both the governmental approach to and social debate on the issue. In general the debate on this issue is between civil rights and gay and lesbian activist groups who support same-sex marriage, and conservative and religious groups who oppose it.

B.5.3 Immigration

Position statement: Immigration into Australia should be maintained or increased because its benefits outweigh any negatives.

Debate over the issue is divided over many sections of politics and society. Those in favour of immigration generally claim that it is good for the economy and that diversity brings cultural enrichment. Those against immigration claim, amongst other issues, that jobs are being taken away from Australians by migrant workers and that cultural homogeneity is more stable.

B.5.4 Republic vs. Monarchy

Position statement: Australia should cease to be a monarchy with the Queen as head of state and become a republic with an Australian head of state.

This is an issue which is divisive amongst political groups and Australians in general. The reasons for support range widely from practical considerations of a need to change (or retain) Australia's political system, to less practical considerations of National identity, to a simple like or dislike of the British monarchy and Australia's ties to it.

B.5.5 WorkChoices

Position statement: Australia should introduce WorkChoices to give employers more control over wages and conditions.

This question was important in Australian politics in the years preceding 2007. In 2007 the issue was decided electorally when the Liberal party, who supported the proposal, were defeated. While this legislation was a topic of debate, unions, civil rights groups and the Labor party were against it, claiming that it infringed on workers rights. The Liberal party and many businesses were in favour, claiming that it would stimulate economic growth and bring great benefits to small businesses.

B.5.6 Carbon tax

Position statement: Australia should introduce a tax on carbon or an emissions trading scheme to combat global warming.

Supporters of the carbon tax include environmental groups and progressives, while those opposed generally claim that it would be bad for Australia's global competitiveness.

B.5.7 Reconciliation

Position statement: The Australian government should formally apologise to the Aboriginal people for past injustices.

Around the time of the civil rights movement in the United States, Australia underwent a similar transformation, with more rights for Australia's original inhabitants. Despite these changes, there were calls for the federal government to issue a formal apology for past atrocities and injustices. The apology was made in 2007, however in the two decades leading up to the apology there was a debate, with conservatives (primarily the Liberal party) arguing that it was unnecessary, while progressives (primarily the Labor party) argued for the apology. Proponents of the apology were sometimes said to have "a black arm-band" view of history, which was considered derisive.

Bibliography

Ahmed Abbasi, Hsinchun Chen, and Arab Salem. 2008. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems*, 26(3):1–34.

Amjad Abu-Jbara, Mona Diab, Pradeep Dasigi, and Dragomir Radev. 2012. Subgroup detection in ideological discussions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 399–409.

Amjad Abu-Jbara and Dragomir Radev. 2012. Subgroup detector: A system for detecting subgroups in online discussions. In *Proceedings of the ACL 2012 System Demonstrations*, pages 133–138.

Lada A. Adamic and Natalie Glance. 2005. The political blogosphere and the 2004 U.S. election: Divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, pages 36–43.

Rakesh Agrawal, Sridhar Rajagopalan, Ramakrishnan Srikant, and Yirong Xu. 2003. Mining newsgroups using networks arising from social behavior. In *Proceedings of the 12th International World Wide Web Conference*, pages 529–535.

Pranav Anand, Marilyn Walker, Rob Abbott, Jean E Fox Tree, Robeson Bowmani, and Michael Minor. 2011. Cats rule and dogs drool!: Classifying stance in online debate. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 1–9.

Alina Andreevskaia and Sabine Bergler. 2006. Mining WordNet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Seventh Conference on International Language Resources and Evaluation*.

Beata Beigman Klebanov, Eyal Beigman, and Daniel Diermeier. 2010. Vocabulary choice as an indicator of perspective. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 253–257. Association for Computational Linguistics.

Sabine Bergler. 1992. *Evidential analysis of reported speech*. Ph.D. thesis.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128.

Rebecca Bruce and Janyce Wiebe. 1999. Recognizing subjectivity: A case study in manual tagging. *Natural Language Engineering*, 5(2):187–205.

Yejin Choi, Eric Breck, and Claire Cardie. 2006. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 431–439.

Yejin Choi and Claire Cardie. 2009. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 590–598.

James R Curran. 2004. *From distributional to semantic similarity*. Ph.D. thesis, University of Edinburgh.

James R. Curran and Stephen Clark. 2003. Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of the Tenth Conference of the European Chapter of the Association for Computational Linguistics*, pages 91–98.

Hoa Trang Dang and Jimmy Lin. 2007. Different structures for evaluating answers to complex questions: Pyramids won't topple, and neither will human assessors. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 768–775.

Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the TAC 2008 opinion question answering and summarization tasks. In *Proceedings of the First Text Analysis Conference*.

Kushal Dave, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International World Wide Web Conference*, pages 519–528.

Peter T. Davis, David Elson, and Judith Klavans. 2003. Methods for precise named entity matching in digital collections. In *Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital libraries*, pages 125–127.

Tim Dawborn and James R. Curran. 2014. Docrep.

Eric de La Clergerie, Benoit Sagot, Rosa Stern, Pascal Denis, Gaelle Recource, and Victor Mignot. 2011. Extracting and visualizing quotations from news wires. *Human Language Technology. Challenges for Computer Science and Linguistics*, pages 522–532.

Pascal Denis and Jason Baldridge. 2008. Specialized models and ranking for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 660–669. Association for Computational Linguistics.

M. Efron. 2004. Cultural orientation: Classifying subjective documents by cociation [sic] analysis. In *Proceedings of the AAAI Fall Symposium on Style and Meaning in Language, Art, Music, and Design*, pages 41–48.

David Elson and Kathleen McKeown. 2010. Automatic attribution of quoted speech in literary narrative. In *Proceedings of the Twenty-Fourth Conference of the Association for the Advancement of Artificial Intelligence*, pages 1013–1019.

Andrea Esuli and Fabrizio Sebastiani. 2005. Determining the semantic orientation of terms through gloss classification. In *Proceedings of the ACM SIGIR Conference on Information and Knowledge Management 2005*, pages 617–624.

Andrea Esuli and Fabrizio Sebastiani. 2006a. Determining term subjectivity and term orientation for opinion mining. In *Proceedings the 11th Meeting of the European Chapter of the Association for Computational Linguistics*, pages 193–200.

Andrea Esuli and Fabrizio Sebastiani. 2006b. SentiWordNet: a publicly available lexical resource for opinion mining. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 417–422.

Andrea Esuli and Fabrizio Sebastiani. 2007a. Pageranking wordnet synsets: An application to opinion mining. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 424–431.

Andrea Esuli and Fabrizio Sebastiani. 2007b. Random-walk models of term semantics: An application to opinion-related properties. *Proceedings of the 3rd Language and Technology Conference*.

Benoit Favre and Dilek Hakkani-Tür. 2009. Phrase-level and word-level strategies for detecting appositions in speech. In *Proceedings of Interspeech*, pages 2711–2714.

Christine Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT press Cambridge, MA.

William Paulo Ducca Fernandes, Eduardo Motta, and Ruy Luiz Milidiú. 2011. Quotation extraction for portuguese. In *Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology*, pages 204–208.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370.

Daniel Gayo-Avello, Panagiotis Metaxas, and Eni Mustafaraj. 2011. Limits of electoral predictions using social media data. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, pages 490–493.

Kevin Glass and Shaun Bangay. 2007. A naïve salience-based method for speaker identification in fiction books. In *Proceedings of the 18th Annual Symposium of the Pattern Recognition Association of South Africa*, pages 1–6.

Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. 2013. Evaluating entity linking with Wikipedia. *Artificial Intelligence*, 194:130–150.

Ahmed Hassan, Amjad Abu-Jbara, and Dragomir Radev. 2012. Detecting subgroups in online discussions by modeling positive and negative relations among participants. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 59–70.

Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 174–181.

Vasileios Hatzivassiloglou and Janyce M. Wiebe. 2000. Effects of adjective orientation and gradability on sentence subjectivity. In *The 18th International Conference on Computational Linguistics*, pages 299–305.

Hua He, Denilson Barbosa, and Grzegorz Kondrak. 2013. Identification of speakers in novels. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1312–1320.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177.

Richard Johansson and Alessandro Moschitti. 2012. Relational features in fine-grained opinion analysis. *Computation Linguistics*, 39(3).

Jaap Kamps, Maartens Marx, Robert J. Mokken, and Maarten De Rijke. 2004. Using WordNet to measure semantic orientation of adjectives. In *Proceedings of the 4th International Conference on Language Resources and Evaluation 2004*, pages 1115–1118.

H. Kanayama and T. Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 355–363.

S. M Kim and E. Hovy. 2006a. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the ACL Workshop on Sentiment and Subjectivity in Text*, pages 1–8.

Soo-Min Kim and Eduard Hovy. 2005. Identifying opinion holders for question answering in opinion texts. In *Proceedings of the AAAI-05 Workshop on Question Answering in Restricted Domains*, pages 1367–1373.

Soo-Min Kim and Eduard Hovy. 2006b. Automatic identification of pro and con reasons in online reviews. In *Proceedings of the COLING/ACL Poster Sessions*, pages 483–490.

Sou-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, page 1367.

Dan Klein and Christopher D Manning. 2002. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems*, pages 3–10.

Moshe Koppel and Jonathan Schler. 2006. The importance of neutral examples for learning sentiment. *Computational Intelligence*, 22(2):100–109.

Ralf Krestel, Sabine Bergler, and René Witte. 2008. Minding the source: Automatic tagging of reported speech in newspaper articles. In *Proceedings of the Sixth International Language Resources and Evaluation*.

Jonathan K. Kummerfeld and Dan Klein. 2013. Error-driven analysis of challenges in coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 265–277.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning*, pages 282–289.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34.

Fangtao Li, Zhicheng Zheng, Tang Yang, Fan Bu, Rong Ge, , Xian Zhang, Xiaoyan Zhu, and Minlie Huang. 2008. THU QUANTA at TAC 2008 QA and RTE track. In *Proccedings of the First Text Analysis Conference*.

Jisheng Liang, Navdeep Dhillon, and Krzysztof Koperski. 2010. A large-scale system for annotating and querying quotations in news feeds. In *Proceedings of the 3rd International Semantic Search Workshop*, pages 1–5.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational linguistics*, pages 768–774.

Robert Malouf and Tony Mullen. 2008. Taking sides: User classification for informal online political discourse. *Internet Research*, 18(2):177–190.

Nuno Mamede and Pedro Chaleira. 2004. Character identification in children stories. *Advances in Natural Language Processing*, pages 82–90.

Yi Mao and Guy Lebanon. 2006. Isotonic conditional random fields and local sentiment flow. In *Advances in Neural Information Processing Systems*, pages 961–968.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330.

Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 432–439.

Panagiotis Takis Metaxas, Eni Mustafaraj, and Daniel Gayo-Avello. 2011. How (not) to predict elections. In *Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on Social Computing*, pages 165–171.

Karo Moilanen and Stephen Pulman. 2008. The good, the bad, and the unknown: Morphosyllabic sentiment tagging of unseen words. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 109–112.

Tony Mullen and Robert Malouf. 2006. A preliminary investigation into sentiment analysis of informal political discourse. In *AAAI Symposium on Computational Approaches to Analysing Weblogs*, pages 159–162.

Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. 2010. Dependency tree-based sentiment classification using CRFs with hidden variables. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 786–794.

Naoaki Okazaki. 2007. CRFsuite: a fast implementation of Conditional Random Fields (CRFs). URL http://www.chokkan.org/software/crfsuite/.

Tim O'Keefe, Silvia Pareti, James R. Curran, Irena Koprinska, and Matthew Honnibal. 2012. A sequence labelling approach to quote attribution. In *Proceedings of the 2012 Joint*

Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 790–799.

Bo Pang and Lillian Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 271–278.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 115–124.

Bo Pang and Lillian Lee. 2008. *Opinion Mining and Sentiment Analysis*. Now Publishers.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 79–86.

Silvia Pareti. 2011. Annotating attribution relations and their features. In *Proceedings of the fourth workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 19–20. ACM.

Silvia Pareti. 2012. A database of attribution relations. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, pages 3213–3217.

Silvia Pareti, Tim O'Keefe, Ioannis Konstas, James R. Curran, and Irena Koprinska. 2013. Automatically detecting and attributing indirect quotations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Souneil Park, Kyung Soon Lee, and Junehwa Song. 2011. Contrasting opposing views of news articles on contentious issues. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 340–349.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Marco Pennacchiotti and Ana-Maria Popescu. 2011. Democrats, Republicans and Starbucks afficionados: User classification in twitter. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 430–438.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of the 2005 Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 339–346.

Bruno Pouliquen, Ralf Steinberger, and Clive Best. 2007. Automatic detection of quotations in multilingual news. In *Proceedings of Recent Advances in Natural Language Processing*, pages 487–492.

Rashmi Prasad, Eleni Miltsakaki, Nikhil Dinesh, Alan Lee, Aravind Joshi, Livio Robaldo, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0 annotation manual. In *Technical report, University of Pennsylvania: Institute for Research in Cognitive Science*.

Will Radford, Will Cannings, Andrew Naoum, Joel Nothman, Glen Pink, Daniel Tse, and James R. Curran. 2013. (Almost) total recall – SYDNEY_CMCRC at TAC 2012. *Proceedings of the Text Analysis Conference 2012*.

Will Radford and James R. Curran. 2013. Joint apposition extraction with syntactic and semantic constraints. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 671–677.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501.

Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta. 2010. Classifying latent user attributes in twitter. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pages 37–44.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142.

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1044–1049.

Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 105–112.

Luis Sarmento and Sergio Nunes. 2009. Automatic extraction of quotes and topics from news feeds. In *4th Doctoral Symposium on Informatics Engineering*.

Nathan Schneider, Rebecca Hwa, Philip Gianfortoni, Dipanjan Das, Michael Heilman, Alan W. Black, Frederik L. Crabbe, and Noah A. Smith. 2010. Visualizing topical quotations over time to understand news discourse. Technical report, Carnegie Mellon University.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Swapna Somasundaran. 2010. *Discourse-level Relations for Opinion Analysis*. Ph.D. thesis, University of Pittsburgh.

Swapna Somasundaran, Galileo Namata, Lise Getoor, and Janyce Wiebe. 2009a. Opinion graphs for polarity and discourse classification. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 66–74.

Swapna Somasundaran, Galileo Namata, Janyce Wiebe, and Lise Getoor. 2009b. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 170–179.

Swapna Somasundaran, Josef Ruppenhofer, and Janyce Wiebe. 2008a. Discourse level opinion relations: An annotation study. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, pages 129–137.

Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing stances in online debates. 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 226–234.

Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological online debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124.

Swapna Somasundaran, Janyce Wiebe, and Josef Ruppenhofer. 2008b. Discourse level opinion interpretation. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 801–808.

Valentin Spitkovsky and Angel Chang. 2012. A cross-lingual dictionary for english wikipedia concepts. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pages 3168–3175.

Veselin Stoyanov, Claire Cardie, Nathan Gilbert, Ellen Riloff, David Buttler, and David Hysom. 2010. Coreference resolution with reconcile. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 156–161.

Veselin Stoyanov, Claire Cardie, Diane Litman, and Janyce Wiebe. 2004. Evaluating an opinion annotation scheme using a new multi-perspective question and answer corpus. In *AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, pages 77–91.

Veselin Stoyanov, Claire Cardie, and Janyce Wiebe. 2005. Multi-perspective question answering using the opqa corpus. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 923–930.

Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2009. Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. 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 656–664.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 327–335.

Ivan Titov and Ryan McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International World Wide Web Conference*, pages 111–120.

Peter Turney. 2002. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424.

Peter D Turney and Michael L Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346.

Vasudeva Varma, Prasad Pingali, Rahul Katragadda, Sai Krishna, Surya Ganesh, Kiran Sarvabhotla, Harish Garapati, Hareen Gopisetty, Vijay Bharath Reddy, Kranthi Reddy, Praveen Bysani, and Rohit Bharadwaj. 2008. IIIT Hyderabad at TAC 2008. In *Proceedings of the First Text Analysis Conference*.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Conference on Message Understanding*, pages 45–52. Association for Computational Linguistics.

Marilyn Walker, Jean Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A corpus for research on deliberation and debate. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pages 812–817.

Ralph Weischedel and Ada Brunstein. 2005. BBN pronoun coreference and entity type corpus. *Linguistic Data Consortium, Philadelphia*.

Stéphanie Weiser and Patrick Watrin. 2012. Extraction of unmarked quotations in newspapers. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*.

Janyce Wiebe. 1990. Identifying subjective characters in narrative. In *Proceedings of the 13th Conference on Computational Linguistics*, pages 401–406.

Janyce Wiebe. 1994. Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287.

Janyce Wiebe. 2000. Learning subjective adjectives from corpora. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 735–740.

Janyce Wiebe, Eric Breck, Chris Buckley, Claire Cardie, Paul Davis, Bruce Fraser, Diane J Litman, David R Pierce, Ellen Riloff, Theresa Wilson, et al. 2003. Recognizing and organizing opinions expressed in the world press. In *AAAI Spring Symposium on New Directions in Question Answering*, pages 12–19.

Janyce Wiebe and Rebecca Bruce. 1995. Probabilistic classifiers for tracking point of view. *Proceedings of the AAAI Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 181–187.

Janyce Wiebe, Rebecca Bruce, and Thomas O'Hara. 1999. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 246–253.

Janyce Wiebe and Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *Computational Linguistics and Intelligent Text Processing*, pages 486–497.

Janyce Wiebe, Theresa Wilson, and Matthew Bell. 2001. Identifying collocations for recognizing opinions. In *Proceedings of the ACL Workshop on Collocation: Computational Extraction, Analysis, and Exploitation*, pages 24–31.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2–3):165–210.

Theresa Wilson. 2008. *Fine-Grained Subjectivity and Sentiment Analysis: Recognizing the Intensity, Polarity, and Attitudes of Private States*. Ph.D. thesis, University of Pittsburgh.

Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005a. OpinionFinder: a system for subjectivity analysis. In *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*, pages 34–35.

Theresa Wilson and Janyce Wiebe. 2003. Annotating opinions in the world press. In *4th SIGdial Workshop on Discourse and Dialogue*, pages 13–22.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005b. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354.

Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of the Third IEEE International Conference on Data Mining*, pages 427–434.

Bei Yu, Stefan Kaufmann, and Daniel Diermeier. 2008. Exploring the characteristics of opinion expressions for political opinion classification. In *Proceedings of the 2008 International Conference on Digital Government Research*, pages 82–91. Montreal, Canada.

Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 129–136.

Jason Zhang, Alan Black, and Richard Sproat. 2003. Identifying speakers in children's stories for speech synthesis. In *Proceedings of the 8th European Conference on Speech Communication and Technology*, pages 2041–2044.