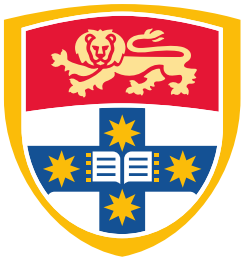
THE UNIVERSITY OF
SYDNEY

Linking Named Entities to Wikipedia

Will Radford

Supervisor: Dr. James R. 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

2015

Abstract

Natural language is fraught with problems of ambiguity, including name reference. A name in text can refer to multiple entities just as an entity can be known by different names. This thesis examines how a *mention* in text can be linked to an external knowledge base (KB), in our case, Wikipedia. The *named entity linking* (NEL) task requires systems to identify the KB entry, or Wikipedia article, that a mention refers to; or, if the KB does not contain the correct entry, return NIL.

Entity linking systems can be complex and we present a framework for analysing their different components. First, mentions must be *extracted* from the text. The KB is *searched* to build a list of candidate entries for a mention. Finally, a *disambiguation* component will identify the correct entry or propose a NIL link. This provides a lens through which to understand and compare systems, and a way to characterise how performance in one component affects another. We use this framework to comprehensively analyse three seminal systems: Bunescu and Paşca (2006), Cucerzan (2007) and Varma et al. (2009). These are evaluated on a common dataset and we show the importance of precise search for linking.

The Text Analysis Conference (TAC) is a major venue for NEL research. We report on our submissions to the entity linking shared task in 2010, 2011 and 2012. Our systems have evolved with the task and we present a state-of-the-art linking system.

The information required to disambiguate entities is often found in the text, close to the mention. We explore apposition, a common way for authors to provide information about entities. We model syntactic and semantic restrictions with a joint model that achieves state-of-the-art apposition extraction performance.

We attempt to use appositions to improve linking with poor results. We generalise from apposition to examine *local descriptions* specified close to the mention. We catalogue how this is used in a recent TAC dataset, showing that entities are described with a variety of attributes using different syntactic mechanisms dependent on their entity type. Moreover, the analysis suggests that KB and NIL mentions are described in

different ways. We add local description to our state-of-the-art linker by using patterns to extract the descriptions and matching against this restricted context. Not only does this make for a more precise match, we are also able to model *failure* to match. Local descriptions help disambiguate entities, further improving our state-of-the-art linker.

The work in this thesis seeks to link textual entity mentions to knowledge bases. Linking is important for any task where external world knowledge is used and resolving ambiguity is fundamental to advancing research into these problems.

Acknowledgements

There are many who deserve credit for this thesis; research is, for me, a social activity and so there are many people to thank. Foremost is James Curran, who is a fantastic supervisor; the last five years has been intellectually stimulating, rewarding and fun. I hope that I have managed to absorb just some of his dedication, thoroughness and taste in research. Ben Hachey has gently shepherded my research down interesting paths with good grace and sense of humour. To find one mentor is lucky, to find a second is a true privilege.

My three examiners have also helped shape this thesis, so I thank Silviu Cucerzan, Dan Weld and Diego Mollá Aliod for their time, effort and care.

Maria Milosavljevic had the uncanny sense of timing to suggest that I return to academic research, just as I had been wondering whether such a thing was possible. Thanks should also to Mike Aitken at the Capital Markets Cooperative Research Centre for providing support and insisting that the research have a foot in both academic and industrial camps. George Wright at Fairfax Media has worked tirelessly to make the Computable News project a vehicle for sneaking our research into production and the public eye.

I have also been fortunate enough to work with a group of friends and colleagues in ə-lab: Daniel Tse, Matt Honnibal, David Vadas, Jonathan Kummerfeld, Stephen Merity, Kellie Webster, Nicky Ringland, Tim O'Keefe, Dominick Ng, Andrew Naoum, James Constable, Richard Billingsley and Candice Loxley. I suspect that all of you have played at least some role in this research, whether working on shared task submissions, generously reading paper drafts or making diplomatic comments about

undercooked practice presentations. Thank you all for the opportunity to learn new things, eat, drink, play and laugh. Thanks especially to Glen Pink for listening to me moan without complaining, on and off the tennis court; Tim Dawborn and Will Cannings for selflessly making machines and software work; and Joel Nothman for your furious intellect and effervescent personality.

I have benefited from wider academic community, thanks to: Jon Patrick, Alan Fekete and Tara Murphy at the University of Sydney for first starts and wise heads in the department; Robert Dale and Mark Johnson at Macquarie University, and Miles Osborne at Edinburgh for good advice; Bonnie Webber for hosting me at Edinburgh and useful conversations about discourse and local description; Stephen Clark, Phil Blunsom, Sophia Ananiadou and Trevor Cohn for their hospitality and hosting the talks that were so critical in developing *how* I communicate my work; finally Mike White for a fresh appreciation of the linguistics of punctuation and the physics of grass-court tennis.

My family have been unwavering in their support. I thank my parents, Mark and Frances, and brothers, James and Nick—it all helped. Finally, I thank my long-suffering wife Kylie and two daughters, Iris and Laura. I am indebted more than I can possibly repay.

Will Radford

26th February 2015

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: *Will Radford*

Signature: Date: *26th February 2015*

Contents

1 Introduction **3**

1.1 Contributions 7

1.1.1 Publications based on this thesis 8

2 Background **9**

2.1 Disambiguating mentions 11

2.1.1 Coreference resolution 12

2.1.2 Cross document coreference resolution 14

2.2 Linking to knowledge bases 16

2.2.1 Linking named entities to Wikipedia 19

2.2.2 Wikification 22

2.3 Entity linking at the Text Analysis Conference 26

2.3.1 2009 28

2.3.2 2010 30

2.3.3 2011 32

2.3.4 2012 35

2.3.5 2013 37

2.3.6 Beyond TAC 38

2.4 Beyond linking text to Wikipedia 42

2.4.1 Linking structured sources 42

2.4.2 Linking streaming data 43

2.4.3 Linking to index 44

2.4.4 Linking to other KBs . . . 45
2.5 Applications . . . 46
2.6 Summary . . . 48

3 Evaluating entity linking **49**
3.1 Cross-document coreference datasets . . . 50
3.2 TAC datasets . . . 53
3.3 Other datasets . . . 57
3.4 Evaluation metrics . . . 59
3.5 Developing a whole-document NEL dataset . . . 62
3.6 Summary . . . 66

4 Benchmarking seminal NEL systems **67**
4.1 A framework for linker analysis . . . 68
4.1.1 Bunescu and Paşca (2006) . . . 70
4.1.2 Cucerzan (2007) . . . 72
4.1.3 Varma et al. (2009) . . . 74
4.2 Evaluation . . . 75
4.3 Error analysis . . . 79
4.3.1 Analysing alias sources . . . 79
4.3.2 Searcher performance . . . 82
4.3.3 The impact of extractors on search . . . 83
4.3.4 Searcher query limits . . . 83
4.3.5 Search errors . . . 84
4.3.6 Effect of extractors on disambiguation . . . 86
4.3.7 Effect of searchers on disambiguation . . . 87
4.3.8 Effect of swapping searchers . . . 88
4.3.9 Disambiguator errors . . . 89
4.4 Summary . . . 91

5 TAC named entity linking **93**

5.0.1 Task description . . . 94

5.0.2 Design principles . . . 96

5.1 The internals of a named entity linker . . . 99

5.1.1 Resources . . . 99

5.1.2 Extraction . . . 105

5.1.3 Search . . . 107

5.1.4 Disambiguation . . . 110

5.1.5 Features . . . 111

5.1.6 NIL clustering . . . 120

5.2 Performance in TAC competition . . . 121

5.2.1 2010 . . . 122

5.2.2 2011 . . . 124

5.2.3 2012 . . . 126

5.3 A state-of-the-art linking system . . . 128

5.3.1 Results on TAC datasets . . . 131

5.3.2 Tuning the NIL threshold . . . 132

5.3.3 Feature analysis . . . 133

5.3.4 Error distribution . . . 135

5.3.5 End-to-end linking . . . 136

5.4 Discussion . . . 139

5.5 Summary . . . 141

6 Extracting Apposition **143**

6.1 Apposition . . . 144

6.1.1 Defining apposition . . . 144

6.1.2 Apposition extraction as a component . . . 148

6.1.3 Evaluating apposition extraction . . . 149

6.2 Apposition in OntoNotes 4 . . . 150

6.2.1 Analysis . . . 151
6.3 Techniques for extracting apposition . . . 154
6.3.1 Syntactic restrictions for phrase candidate selection . . . 154
6.3.2 Semantic compatibility features . . . 155
6.3.3 Joint classification . . . 157
6.4 Apposition extraction systems . . . 158
6.4.1 Pattern . . . 158
6.4.2 Adjacent NPs . . . 160
6.4.3 Rule . . . 160
6.4.4 Labelled Berkeley Parser . . . 162
6.4.5 Phrase classification . . . 163
6.4.6 Joint classification . . . 163
6.5 Results . . . 164
6.6 Summary . . . 171

7 Local description for NEL **173**
7.1 Apposition for NEL . . . 174
7.2 Describing entities . . . 176
7.3 An analysis of local description . . . 178
7.3.1 Locations . . . 178
7.3.2 Organisations . . . 181
7.3.3 People . . . 183
7.4 Local description in TAC 11 . . . 185
7.5 Extracting local description . . . 189
7.6 Linking with local description . . . 194
7.6.1 Results . . . 197
7.6.2 Feature weight analysis . . . 201
7.7 Summary . . . 204

8 Conclusion **207**

8.1 Future work 208

8.2 Summary 211

Bibliography **213**

List of Figures

2.1 Excerpt from "John Howard warns Liberals frail Labor will rise again". 12

2.2 Excerpt from the Wikipedia article and categories for `John Howard`. . . 19

2.3 Excerpt from the Wikipedia article for `John Howard (disambiguation)`. 20

2.4 Excerpt from the Wikipedia article's infobox for `John Howard`. 20

3.1 Example TAC query and excerpt from its document. 53

3.2 Excerpt from the TAC KB entry for Bud Abbott. 54

3.3 Example of B^{3+}. 60

4.1 Effect of query limit on searcher candidate recall. 84

5.1 Named Entity Linking system diagram. 99

5.2 Operations mapping KB_i to KB_j: remove, rename, split and merge. . . 102

5.3 Link graph reweighting sketch. 114

5.4 Ideal NIL threshold for the unsupervised NEL system ($\mathcal{U}$) on TAC 09 . 132

6.1 Appositional NP candidates. 154

6.2 Non-appositional NP candidates. 155

6.3 Non-appositional but restricted candidates. 155

6.4 Example single phrase classification, with HEAD and ATTR the two labels. 157

6.5 Joint classification, with HEAD-ATTR the label. 157

6.6 Labelled appositional NP candidates. 162

7.1 Location description rules. 191

7.2 Organisation description rules. 192
7.3 Person description extraction performance and rules. 193

List of Tables

2.1 Continuum of structured information. 17
2.2 Results from TAC 09. 29
2.3 Results from TAC 10 without web access. 31
2.4 Results from TAC 11 without web access. 33
2.5 Results from TAC 12 without web access. 35
2.6 Results from TAC 13 without web access. 38
2.7 Results outside TAC competition. 39

3.1 Summary of NEL datasets. 50
3.2 Statistics for the TAC datasets. 55
3.3 Ambiguity in the TAC datasets. 56
3.4 Match statistics for KB queries. 57
3.5 Notation for searcher analysis measures. 59
3.6 SMH corpus entity annotation scheme. 63
3.7 SMH dataset statistics. 64
3.8 SMH dataset ambiguity. 64
3.9 SMH dataset match statistics for KB mentions. 65

4.1 Comparative summary of seminal linkers. 70
4.2 Results on TAC 09 for baselines, systems and literature. 76
4.3 Result on TAC 10 for baselines, systems and literature. 76
4.4 Results on TAC 10 genre and entity type subsets. 78
4.5 Search over individual alias fields (TAC 09). 80

4.6 Search over multiple alias fields (TAC 09). 80
4.7 Backoff search over alias fields (TAC 09). 81
4.8 Performance of searchers from the literature (TAC 09). 82
4.9 Effect of coreference/acronym handling on searching (TAC 09). 83
4.10 Number of KB accuracy errors due to search (TAC 09). 85
4.11 Distribution of searcher errors on TAC 09 queries 86
4.12 Coreference analysis over 100 queries sampled from the TAC 09. . . . 86
4.13 Effect of coreference/acronym handling on linking (TAC 09). 87
4.14 Effect of searchers on cosine disambiguation (TAC 09). 88
4.15 Combinations of searchers on implemented disambiguators (TAC 09). 89
4.16 Number of KB accuracy errors due to disambiguation. 89
4.17 Distribution of disambiguator errors on TAC 09 queries 90
4.18 Characteristic errors over TAC 09 queries 90

5.1 Linking terminology 96
5.2 Overview of our TAC systems. 100
5.3 Information from different fields of the Wikipedia article `John Howard`. 101
5.4 Statistics mapping from the TAC KB to our 2012 Wikipedia snapshot. . 103
5.5 Tuned search fields. 109
5.6 Terminology used for Cucerzan (2007) features. 112
5.7 Wikipedia article graph terminology. 113
5.8 Features used in our TAC 12 linkers. 120
5.9 Overview of TAC results in 2010, 2011 and 2012. Bold results indicate the figure for a particular metric of our systems. 123
5.10 Accuracies on TAC 10-eval by genre and entity type. 124
5.11 B^{3+} F scores on TAC 11 by genre and entity type. 125
5.12 Newswire B^{3+} F scores for 11.3 by query type and entity type. 125
5.13 B^{3+} F score on TAC 12 by genre and entity type. 127
5.14 Configuration for the unsupervised NEL system ($\mathcal{U}$). 130

5.15 Configuration for the supervised NEL systems ($\mathcal{S}$ and $\mathcal{S}+$). 130
5.16 Results on TAC datasets with current linkers. 131
5.17 Threshold analysis. 133
5.18 Feature ablation using the unsupervised NEL system ($\mathcal{U}$). 134
5.19 Feature ablation using the supervised NEL system ($\mathcal{S}$). 135
5.20 Error profiles on TAC datasets with current linkers 137
5.21 Performance on the SMH DEV dataset. 137
5.22 Performance on the SMH TEST dataset. 138

6.1 Syntactic distribution of apposition in the Brown corpus. 146
6.2 Genre analysis in OntoNotes 4. 151
6.3 Sentence and apposition distribution. 151
6.4 Apposition forms in TRAIN. 152
6.5 The top-20 HEAD/ATTR NE tag patterns in TRAIN. 153
6.6 Seed WordNet synsets used to create the semantic gazetteers. 156
6.7 The top-five patterns by recall in the TRAIN dataset. 160
6.8 Full set of apposition extraction rules. 161
6.9 Results over DEV. 165
6.10 Results over DEV, ablative analysis. 166
6.11 Results over DEV, with LBP trees and gold resources. 166
6.12 Results for Pattern and Joint LBP on the different domains in DEV. . . 168
6.13 Results over TEST_F with regular and labelled parse trees. 169
6.14 Results over TEST with regular and labelled parse trees. 169
6.15 Selected BP/LBP parse error distribution. 171

7.1 Impact of increasingly local similarity scope for NEL on TAC 11. 175
7.2 Overview of local description types 179
7.3 Distribution of local description in TAC 11 queries. 185
7.4 Distribution of local description by information type 186
7.5 Distribution of LOC local description 187

7.6 Distribution of ORG local description 188
7.7 Distribution of PER local description 189
7.8 Location description performance on TAC 11. 191
7.9 Organisation description performance on TAC 11. 192
7.10 Person description performance on TAC 11. 193
7.11 Features from an org-sponsor/per-left-pos description match. 195
7.12 Features from an org-sponsor/per-left-pos description without a match. 195
7.13 TAC 11 query matches. 197
7.14 TAC 11 field matches. 198
7.15 Local description linking performance. 199
7.16 Linking performance using TAC 11 threshold. 200
7.17 Ablative analysis for $\mathcal{S}$ using the threshold optimised on TAC 11. . . . 201
7.18 Top 10 positive weights for $\mathcal{S}$. 202
7.19 Top 10 negative weights for $\mathcal{S}$. 203
7.20 Results on SMH TEST. Thresholds are optimised on SMH DEV. 203

1 Introduction

My sympathies go out
Go out to John Howard the actor
His nomenclature
Messed up under history's tractor.

Ross McLennan in "John Howard the Actor", 2004

Natural language is an ambiguous medium for communication and while humans unconsciously negotiate and resolve this uncertainty in most instances, automated systems cannot. The ambiguity problem manifests itself in several ways, among them how language refers to entities and general concepts. A *mention* is a phrase that refers to an entity or general concept and is used to link it to a shared context that helps interpret the information in the communication. The two entries below are from Wikipedia, an online encyclopaedia, for two distinct people named John Howard, a former Australian prime minister (Example 1) and an Australian actor (Example 2):

(1) John Winston Howard, OM AC SSI, (born 26 July 1939) is an Australian politician who served as the 25th Prime Minister of Australia, from 11 March 1996 to 3 December 2007.

(2) John Howard (born 22 October 1952 in Corowa, New South Wales) is an Australian stage and screen actor.

As is the case here, names can be ambiguous—John Howard can refer to both entities—and entities can be known by different names—John Winston Howard and John Howard can refer to `John Howard`[1]. Correct linking requires interpreting the mention's context.

Linking names in language to an external knowledge base (KB) benefits both. External knowledge provides context to understand a statement, and information conveyed in a statement enriches and enhances the external knowledge. Moreover, external knowledge and facts can help disambiguate names. The quotation in Example 3 mentions Howard, and the names of television series and films.

(3) Howard is best known for his appearances in the film The Club, and the television series SeaChange, Always Greener, All Saints and Packed To The Rafters.

Related entities are valuable context for disambiguating the mention Howard to `John Howard (Australian Actor)`, but of course some mentions may themselves be ambiguous (All Saints can refer to `All Saints (group)` or `All Saints (TV series)`).

External knowledge can help disambiguation when context is limited. Example 4 (Farrell, SMH 2013-11-06)[2], shows a hyperlinked headline that is displayed in isolation on a news website front page.

(4) Why is Howard anti-science?

Any context that might disambiguate Howard is found in the linked story, but readers familiar with Australian politics or the news website would assume, in the absence of other information, that the mention refers to `John Howard`. A KB can provide statistics about the prominent links for a particular mention and, in this case, Wikipedia's most prominent John Howard is `John Howard`.

As useful as external knowledge is, it can never have complete coverage and lags behind the world that it describes.

[1] The article `John Howard` is about the politician. Other entities named John Howard have Wikipedia article titles with a disambiguating suffix, for example `John Howard (Australian actor)`.

[2] www.smh.com.au/comment/why-is-howard-antiscience

(5) It was handed over fair and square years earlier by an acolyte of John Howard, the Klan leader who founded the shop. . . . Mr. Howard is a notably cantankerous fellow in his mid-60s.

The John Howard in Example 5 (Severson, NYT 2012-01-13)[3] does not match a Wikipedia article and should be linked NIL. Information about him is contained in the text: the Klan leader who founded the shop and Mr. Howard . . . mid-60s. This precise information about his occupation and age may help a reader to deduce that this is a different John Howard, not present in the KB. Resolving name ambiguity and recognising NIL mentions are the key components of the named entity linking task.

Name ambiguity is problematic for many applications that interface structured and unstructured information. We define structured information as any case where it is unambiguously specified, such as a KB record, where information is split into discrete fields. Semi-structured text includes formalised descriptions, for example John Winston Howard, OM AC SSI, (born 26 July 1939) in Example 1. This entity mention is accompanied by given names, honours and birthdate in an order defined by convention. Unstructured information, on the other hand, requires sophisticated interpretation of context to extract and normalise it before use.

(6)

USER	I want to know about Johnny Howard, the former PM.
SYSTEM	`Did you mean John Howard?`
USER	Yes, what is his birthday?
SYSTEM	`26th July 1939.`

Consider a dialogue with a hypothetical KB interface in Example 6. Resolving ambiguity is only part of question answering, but it is necessary for several steps of the solution. The system would need to recognise that Johnny Howard is an alternative name for `John Howard`, and use contextual cues such as the former PM to distinguish between different candidates. Then, the system must identify that the pronoun his refers to the same entity as John Howard. Finally, selecting which fact to return to the

[3] www.nytimes.com/2012/01/13/us/in-laurens-sc-the-redneck-shop-and-its-neighbor

user may require general fact extraction to recognise that birthday corresponds to a fact that might be labelled *date of birth*.

Systems such as the one described above require a grasp of natural language understanding, inference and generation to bridge the gap between unstructured and structured knowledge. While consumer applications such as Apple's *Siri*[4] and Google's *Knowledge Graph* for web search[5] hint at this direction, true open-domain systems are currently still out of reach. However, resolving name ambiguity is still useful for more focused applications.

The Atlantic reported in 2011 (Madrigal, Atlantic 2011-03-11)[6] on curious price movements in Berkshire Hathaway, the influential holding company. They postulate that this was due to algorithmic trading strategies assuming that spikes in news about Hathaway referred to the company rather than Anne Hathaway, the actress who appeared in films opening at the same time. Failure to address name ambiguity may also distort metrics that characterise how entities are related, by merging multiple entities together (Fegley and Torvik, 2013). Extracted and disambiguated entities can also provide insight into large document collections. Indexing documents by entity occurrence allows systems to automatically create entity timelines (Mazeika et al., 2011) and co-occurring entities form graphs to help explore entity relations (Malik et al., 2011; Hossain et al., 2012). A KB can be populated from unstructured data, but is subject to name ambiguity problems. Facts and attributes about an entity may refer to it using an ambiguous name or pronoun, and these facts must be added to the correct KB entry. If the KB is to grow, any NIL mentions should be clustered so that they can form a new entry. The whole process of knowledge base population relies on accurate name ambiguity resolution.

[4] www.apple.com/ios/siri

[5] www.google.com/insidesearch/features/search/knowledge

[6] www.theatlantic.com/technology/archive/2011/03/does-anne-hathaway-news-drive-berkshire-hathaways-stock

1.1 Contributions

This thesis addresses the problem of *Named Entity Linking* (NEL), where named entity mentions must be linked to an external knowledge base—in our case, Wikipedia. Chapter 2 reviews how name ambiguity has been addressed in previous literature and how the combination of matching KB records and recognising NIL entities is a distinct problem. We also describe two key techniques for disambiguation: using the document context and harnessing the KB structure. Chapter 3 outlines some datasets and metrics used to evaluate the performance of NEL systems. Our first contribution is a detailed analysis in Chapter 4 of three different systems from the literature evaluated on a common dataset. Shared tasks are important drivers for Natural Language Processing (NLP) research and NEL is no different. Chapter 5 describes our participation in the Knowledge Base Population (KBP) track of the Text Analysis Conference (TAC) workshop in 2010, 2011 and 2012. Our second contribution is to illustrate how our systems use context similarity and KB structure to disambiguate query entities in a state-of-the-art NEL system.

Existing approaches to NEL take advantage of coarse context at the document, paragraph or sentence level. We argue that accounting for precise entity description is critical for improving performance beyond the strong baselines of existing methods. Our final contributions are some preliminary work towards this goal. We survey existing work in apposition extraction in Chapter 6 and propose systems that take advantage of syntactic and semantic features to achieve state-of-the-art performance. In Chapter 7, we find that, while apposition is a prominent method for specifying entity attributes, it is too infrequent to improve disambiguation itself. We describe the analysis of a TAC dataset that characterises how disambiguating information is specified. We propose manual rules for extracting local description—attributes specified in close context to the mention. We integrate these into our state-of-the-art linking system, analyse their impact and discuss future directions to realise the goal of precise information extraction for disambiguation.

In summary, we present a framework for analysing NEL systems and demonstrate how it applies to seminal systems and has directed our research in developing a state-of-the-art system in the TAC shared task. We have used our framework to explore how entities are introduced and described in text and how this information can be used for disambiguation. The framework and techniques described in this thesis are a strong foundation for anyone wishing to link mentions in text to knowledge bases. Linking entities is important anywhere external world knowledge is used, including: search, question answering and fact extraction. Resolving entity ambiguity is fundamental to advancing research into these applications.

1.1.1 Publications based on this thesis

Parts of this thesis have been reported in conference proceedings and journals. The framework for analysing NEL and analysis of three seminal systems (Chapter 4) appears in Hachey et al. (2013). While not the first author, I was responsible for much of the implementation and analysis, providing substantial parts of the paper text. Our submissions to the TAC shared task, but not the state-of-the-art system, that are presented in Chapter 5 are described in system reports (Radford et al., 2010, 2011, 2012). Work on apposition extraction is reported in Radford and Curran (2013).

2 Background

> Former Labor deputy prime minister Lionel Bowen has died, aged 89. . . . Immigration Minister Chris Bowen, told AAP he was often in the position of telling those who asked that he was not related to the former Labor deputy. "Whenever I said no, . . ." Mr Bowen *[Chris Bowen]* said.
>
> Ambiguous Bowens (Editors, SMH 2012-04-01)[1]

One of the myriad uses of language is to communicate information about abstract or concrete concepts. In Chapter 1, we introduced named entity linking (NEL), the task of linking names in text to external knowledge bases (KBs) and two problems that make it difficult: resolving name ambiguity and recognising NIL mentions. We also distinguished structured information, which is well-specified, from unstructured information. Finally, we motivated how resolving name ambiguity and recognising NIL mentions is important to a wide range of tasks that process natural language.

Having introduced some terms relevant to the tasks, we review them briefly in a more formal manner. Broadly, a *mention* is a phrase or fragment, for our purposes in text, that plays a referring role in the discourse. These may be proper nouns (e.g. John Howard), common nouns (e.g. man of steel) or pronouns (e.g. he). This work focuses on proper noun mentions, or names, that refer to *entities*. These are the people, places, organisations, etc. that feature in the discourse. Named entity linking further distinguishes entities into two classes, by their inclusion or exclusion in an

[1] www.smh.com.au/national/former-deputy-pm-lionel-bowen-dead

external knowledge resource: KB and NIL entities. NEL considers name mentions and entities with respect to a KB, but other tasks have different scope. Coreference resolution (Section 2.1.1) considers all types of mentions. Cross-document coreference resolution (Section 2.1.2) typically addresses names. Neither link to entities in an external KB. Wikification (Section 2.2.2) links names and nominal mentions to an external wiki, typically articles discussing a topic. These articles often describe an entity (e.g. the biography of a person), as in NEL, but can also be general concepts, such as `steel`. In this case, we refer to *concept linking*, where concepts include entities and general concepts.

This chapter reports on how the problem of mention ambiguity has been explored in coreference resolution, both within and between documents or discourses. Approaches to both tasks use mention context, hypothesising that mentions of the same entity occur in similar contexts. We then discuss previous work in linking mentions to KBs—both named entities and general concepts. External knowledge sources provide more context to match entities against, and approaches can take advantage of statistics, facts and structure that exists in the KB. The combination of resolving name ambiguity and NIL recognition differentiates NEL from other related tasks and is explicitly evaluated in the Text Analysis Conference. We review the English Entity Linking task and conclude with discussion of related areas and NEL applications.

There has been substantial recent interest in NEL and this chapter aims to provide an overview of the main approaches, and their context within the field. We defer a detailed comparison of NEL systems until Chapter 4, where we describe the main approaches in detail, and empirically compare and analyse our reimplementation of the main system—the first contribution of this thesis. The metrics and datasets used to evaluate NEL are described in full in Chapter 3, but we introduce them below to help place the results in this chapter in context.

Accuracy over a set of mention is the proportion that are correctly linked. Datasets are usually heterogeneous, with some mentions that should be linked to the KB, and some that should not. A correct response for the former requires the correct

entity to be returned, but a NIL answer suffices for the latter. Accuracy is reported micro-averaged over all queries and often for the KB and NIL subsets alone. When gold-standard clusters exist for NIL queries, they can be used for evaluation. B^{3+} measures how well a system clustering matches the gold standard, with the added restriction that KB queries should be linked correctly. The F score over all queries is most often reported and balances under- and over-clustering, and precision, recall and F can be reported for all, KB or NIL clusters. The proliferation of metrics means that it is not always straightforward to compare systems. All performance figures presented here are as published in the literature, and we occasionally omit figures where a system does not report complete performance.

We assert that entity mentions are often accompanied by local description—precise specification of one or more of an entity's attributes in the same sentence. While existing methods use broad context similarity for disambiguation, making use of more detailed information is important for improving performance in disambiguation and recognising NILs. As such, we highlight where approaches extract and use local description as we believe that ultimately systems should take advantage of the same cues that readers use to understand entities.

2.1 Disambiguating mentions

Ambiguous mentions have been studied in many types of text including news (Wacholder et al., 1997), web documents (Mann and Yarowsky, 2003), academic mailing lists (Hassell et al., 2006) and email collections (Minkov et al., 2006). Ambiguity affects applications that use extracted entities. Ambiguous names and aliases can distort entity network measures (Fegley and Torvik, 2013) by "lumping" entities together. This results in high degree vertices that can be penalised by some measures. Fegley and Torvik also suggest that some power-law observations for collaborator counts are an artefact of name ambiguity. This section describes research into name ambiguity and how mention context is used to disambiguate entities.

2.1.1 Coreference resolution

Mention referents can be ambiguous within a document. Coreference resolution aims to group mentions in the same document that share referents. Figure 2.1 shows an excerpt from a news article containing several entity mentions and a cluster of coreferent mentions. These cover a range of syntactic types: proper noun, common noun and pronoun, and each has an anaphoric relationship with the previous.

> Bill Shorten's beleaguered Labor MPs should not despair at being banished to the electoral wilderness, says no lesser authority on Australian politics than Labor's arch-enemy, John Winston Howard. The fiercely partisan former Liberal prime minister said Labor would bounce back.
>
> He said "Australia's oldest political party" should take heed from history noting that even after the electoral debacle of 1975 when Gough Whitlam's shambolic government was bundled out in a landslide of record proportions, a new Labor state government, led by Neville Wran, was elected in the largest state of NSW within six months.

John Howard (from the headline)
Labor's arch-enemy
John Winston Howard
The fiercely partisan former Liberal prime minister
He

Figure 2.1: Excerpt from "John Howard warns Liberals frail Labor will rise again" (Kenny, SMH 2013-10-22)[2] and a cluster of John Howard mentions.

There have been a wide variety of approaches to coreference resolution, from sophisticated models (Haghighi and Klein, 2010; Chang et al., 2013) to simpler systems (Raghunathan et al., 2010). Most systems focus on a mention's syntactic and semantic characteristics to decide which other mentions it should be clustered with, if any. We focus on systems that extract local information about mentions to help clustering. Haghighi and Klein (2009) view coreference resolution as a process that builds clusters by deciding to cluster (or not) pairs of mentions, taking syntactic and semantic features into account. One clustering constraint requires checking "role appositives" that can

[2] www.smh.com.au/federal-politics/john-howard-warns-liberals-frail-labor-will-rise-again

specify a person entity's profession. This includes apposition such as Labor's arch-enemy, where the benefit is two-fold: the apposition relation mandates that it be clustered with its adjacent noun phrase John Howard and also that it provides more context to cluster with The fiercely partisan former Liberal prime minister.

Raghunathan et al. (2010) present a deterministic system for coreference that uses a sequence of increasingly general sieves to cluster mentions. While their goal is to cluster pronominal, common and proper noun mentions, they attempt to characterise "role appositives" for gender-non-neutral, animate person mentions. They find that their simple coreference models outperform more complex ones on at least two test sets. They attribute their improved performance over Haghighi and Klein (2009) to repeated application of sieves of decreasing precision and a richer feature model—necessary to cluster mentions that are more obliquely related.

Coreference resolution must also account for singleton mentions (e.g. Gough Whitlam in Figure 2.1) and systems should assign them to their own cluster. This may be simple when considering well-specified proper nouns that are not similar to others, but common-noun references may require external knowledge and inference to identify that a Liberal prime minister would be Labor's arch-enemy. Recasens et al. (2013) attempt to classify whether a mention is a singleton and train a logistic regression model that uses morphosyntactic, grammatical and semantic features. Their method performs well in isolation and contributes to better coreference resolution. Singleton recognition is an analogue of NIL recognition as it requires identification of elements that should not be matched. This approach is interesting as it models non-matching explicitly, rather than attempting to match and finding a low similarity.

Li et al. (2004) model how mentions are generated in documents. Their joint model over a document accounts for generation of entities, at least one well-specified exemplar mention of each entity and how exemplars are reformulated to form complete coreference chains. While this does not consider pronominal or nominal mentions, the method rests on trying to interpret the cues that authors use to introduce an entity, using at least one fully-specified mention for a reader to easily resolve.

2.1.2 Cross document coreference resolution

Cross document coreference resolution (CDCR) expands the single document scope to whole collections. This brings scalability considerations to the fore as techniques that are efficient enough for one document may not scale to large corpora. Although moving beyond the single document increases computational complexity, corpora allow more context to be used. In regular coreference resolution, all mentions share a document and context is necessarily local, whereas many CDCR systems use a mention's document as context for disambiguation.

Wacholder et al. (1997) describe heuristics used to disambiguate proper nouns in Wall Street Journal news stories. The *Nominator* system primarily uses document context of co-occurring names, but can incorporate world knowledge in the form of authority lists to match names against. This early work has two key contributions: the use of *whole document* context to disambiguate names and the explicit use of unambiguous names. Context is vital for disambiguation and any name with only one candidate is a fixed point of reference that helps disambiguate other mentions.

Bagga and Baldwin (1998b) use a vector space model (VSM) where a mention's context is represented as term vectors from a mention's sentences. These summary term vectors are compared and those above a similarity threshold are clustered together. They adopt an incremental clustering approach where items are compared with existing clusters and are either added to a similar cluster or create their own. This approach is efficient since it requires only one pass over the mentions, but can be less effective than more inefficient methods that can take the whole space of mentions into account while building clusters, such as hierarchical agglomerative clustering. They evaluate over 197 New York Times (NYT) news articles containing the name John Smith. They also introduce the B^3 metric (Bagga and Baldwin, 1998a), that evaluates how well a clustering matches gold-standard clustering by judging correctness for each mention. This improves upon the link-oriented MUC coreference score, which does not give credit for separating singleton clusters and does not penalise precision

appropriately. B^3 is the basis for the B^{3+} metric used for NEL evaluations, which we cover in depth in Section 3.4.

Several systems attempt to confront the scaling issues inherent in CDCR. Gooi and Allan (2004) introduce a larger corpus—25K person name mentions from the NYT. Their Person X dataset uses a pseudo-name technique to create disambiguation data. One mention (automatically recognised) is randomly chosen from 34,404 documents. Documents containing this mention are inspected and an unambiguous name chosen by an annotator (e.g. if the mention is John Howard, the unambiguous name might be John Winston Howard). Then, the mention is replaced with person-x and the unambiguous name is taken to be the gold-standard cluster label. They use a VSM to cluster entity mentions, not coreference chains, and "snippet", which is a 55-token window centred on the mention that may cross sentence boundaries. They experiment with different clustering approaches and find that hierarchical agglomerative clustering (HAC) performs better than incremental clustering but is noticeably slower.

Rao et al. (2010) use hash functions to select candidate clusters for merging. Their approach takes best-case linear time rather than the quadratic required for HAC and uses lexical and topical similarity and they find similar clustering accuracy to slower methods. Latent Dirichlet Allocation (LDA) topics (Blei et al., 2003) helps account for lexical sparseness and identify more general topics for similarity. Singh et al. (2011) propose a large-scale clustering method that represents CDCR as an undirected graphical model, where assignments are optimised using approximate inference. During computation, new cluster assignments are proposed and the parallel tasks are structured using a hierarchy to optimise assignment efficiency. Experiments on the Person-X corpus (Gooi and Allan, 2004) show that their method reaches the same accuracy as pairwise clustering in 10% of the runtime. On a large-scale corpus of hyperlink anchors to Wikipedia (Singh et al., 2012), their method scores 73.7% F B^3.

Specific attributes can also help disambiguate entities. Mann and Yarowsky (2003) use a combination of manual patterns and bootstrapped templates to extract precise personal information including birth year, occupation, spouse, familial relationships

and nationality. They use web search queries to create datasets for evaluation. The first uses pseudo-names, where pages from two distinct names are retrieved from the Google search engine and those names replaced with the same "false" name. The other uses websites containing naturally ambiguous names (e.g. Jim Clark) that have been manually disambiguated.

They use HAC to cluster these using similarity of term vectors and biographical facts for occupation, birth year, spouse, birth location and school. The biographic facts increase clustering accuracy (i.e. proportion of pages where the system identified the correct cluster) by 3.5% to 86.4% on 28 pseudo-names. They classify mentions from the naturally ambiguous names into three clusters (the two major senses and "other"), which performs worse at 75%–80% accuracy. Attributes can be used to synthesise disambiguating summaries for human consumption. Schiffman et al. (2001) group together appositive phrases that describe personal attributes. They traverse the WordNet (Miller, 1995) hierarchy to find occupation terms, which are used to extract appositive phrases. These phrases are merged with relative clauses to generate summaries.

Clustering web pages with ambiguous person mentions is a core task in the Web Person Search workshop (Artiles et al., 2007, 2009, WEPS), with an extension task to extract a fixed set of personal attributes. Many systems take a VSM clustering approach. For example Rao et al. (2007) use K-means to cluster bags of words (BOW), part-of-speech (POS) tags, entities, occupations and titles. Gong and Oard (2009) explore a key problem with entity clustering—how can systems be prevented from over or under-clustering items. They learn optimal cut-points in HAC using a SVM model over WEPS data.

2.2 Linking to knowledge bases

Coreference resolution approaches have shown how context—broad lexical, topical and local—can disambiguate mentions within documents and at scale. Framed as

Structure	Resource	Data
Low	Sentence	John Howard... The fiercely partisan former Liberal prime minister
	Apposition	John Howard, the former prime minister, ...
Med.	Wiki. article	`[John Winston Howard], OM AC SSI, (born 26 July 1939) is an Australian politician who served as the 25th [Prime Minister] of [Australia]...`
	Wiki. Infobox	`office = [25th] [Prime Minister of Australia]`
High	Freebase	`<John Howard, /government_positions_held, Prime Minister of Australia>`
	Freebase MIDs	`</m/0chh05, /m/02xlhc6, /m/060f2>`

Table 2.1: Continuum of structured information expressing the fact that John Howard was, at one time, the Australian Prime Minister. Square brackets in Wikipedia markup indicate a hyperlink.

a linking task, coreference resolution involves matching like items, for example a mention's context. Linking to an external KB, as motivated in Chapter 1, marks a shift: rather than assessing the similarity between two mention contexts, a system must compare similarity between a mention's context and a KB entry.

We use a broad definition of KB, considering structured collections of resource description framework (RDF) records such as DBpedia[3], YAGO[4] and Freebase[5] as well as densely linked textual resources such as Wikipedia.[6] Table 2.1 shows a continuum of structure used to express that John Howard was, at one time, Prime Minister of Australia. This includes unstructured text, moderately structured encyclopaedic text with highly structured, disambiguated links, and structured tuples. The intelligence of the processing required to extract and interpret the information decreases with the level of structure. Wikipedia article text contains hyperlinks to other articles and its more formal style makes it more straightforward to process. The markup also includes structured templates such as infoboxes that contain key-value pairs of entity

[3] http://dbpedia.org
[4] www.mpi-inf.mpg.de/yago-naga/yago
[5] www.freebase.com
[6] www.wikipedia.org

attributes. Disambiguation depends on whether an author applied markup to a span of text and while they operate under editorial guidelines, the authors can produce noisy or inconsistent results. Highly structured resources use exactly specified and disambiguated records that conform with a schema. This may be rendered in human- and machine-readable form, as in the last two rows of Table 2.1.

The main advantage of linking against a KB is that attributes associated with the entry can provide a larger "surface" to match against. A KB entry that contains biographical text will cover a wide range of events, and a mention from a news article may not. For example, `John Howard` has held different roles in different governments, including Treasurer and Prime Minister, both listed in his Wikipedia article. News stories from different periods may variously describe him as Treasurer John Howard or Prime Minister John Howard, which may have a less than perfect context match. Linking against a Wikipedia-based KB would allow matching to the article, which matches both contexts. As well as greater textual context, a KB can provide entity prominence information and facts. Entity prominence can be estimated from an entity's frequency of reference in a KB. Where little context is available, linking a mention to the most popular matching entity is a suitable strategy, in the same way as the most-frequent-sense baseline is strong in WSD. Facts allow more precise disambiguation as a mention context can be checked for lexical overlap with the fact, or presence of related entities expressed by the fact. More broadly, if we interpret co-occurring entity mentions as related, we might expect this to be reflected as a relationship between their KB entries.

Linking to a KB requires different approaches to CDCR. While both begin with an *extraction* phase where mentions are identified in the text, linking systems must *search* the KB for candidate entries—possible matches. Next, a *disambiguation* step identifies the correct referent. As no KB has complete coverage, a system may identify that a mention should not be linked to any KB node, known as a NIL link. At this point, CDCR approaches can be used to cluster NILs that refer to the same entity. Adding NIL clusters to the KB, coupled with fact extraction, is a way to automatically populate KBs from text.

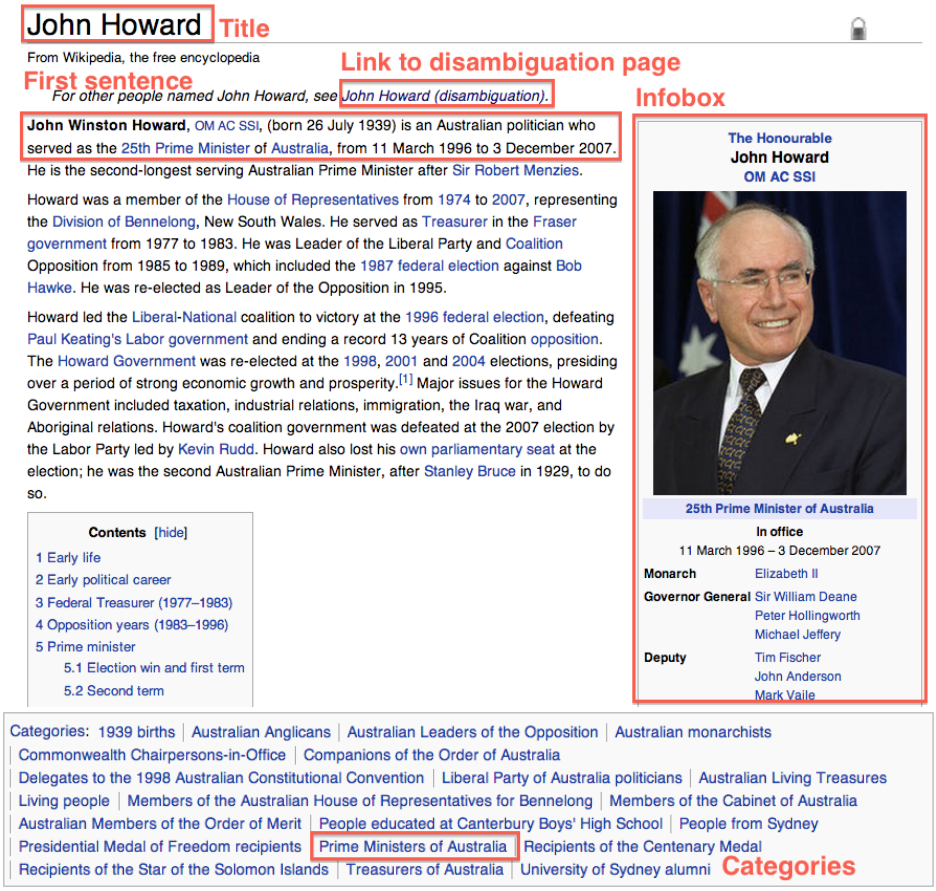


Figure 2.2: Excerpt from the Wikipedia article and categories for `John Howard`.

2.2.1 Linking named entities to Wikipedia

Wikipedia[7] is a web-scale, collaboratively edited encyclopaedia. English Wikipedia is the largest of many language editions at 4.3M articles[8], the majority of which are entities. Volunteer editors effectively curate the KB, applying style guidelines and inclusion rules that require articles to be "notable".

The KB structure has a number of useful reference features: *redirect* articles provide alternative names and *disambiguation* (see Figure 2.3) pages collect and explicitly list ambiguous entities for a name. The articles (see Figure 2.2) can contain structured data in *infoboxes* (see Figure 2.4), can be tagged with topical *categories* and contain

[7] www.wikipedia.org
[8] As of 3rd November 2013

John Howard (disambiguation)

From Wikipedia, the free encyclopedia

John Howard, Prime Minister of Australia from 1996 to 2007.

John Howard may also refer to:

Contents [hide]

1 Other politicians
2 Actors
3 Sports
4 Architects
5 Musicians
6 Others
7 See also

Other politicians [edit]

- John Howard (died 1437), MP for Essex, Cambridgeshire and Suffolk
- John Howard (MP for Faversham) (1863–1911), British Member of Parliament for Faversham 1900–1906
- John Howard (MP for Southampton) (1913–1982), British Member of Parliament, 1955–1964
- John Eager Howard (1752–1827), U.S. Senator from Maryland
- John Morgan Howard (1837–1891), British judge and Conservative Party politician

Actors [edit]

- John Howard (Australian actor) (born 1952), Australian actor
- John Howard (American actor) (1913–1995), American actor

Figure 2.3: Excerpt from the Wikipedia article for `John Howard (disambiguation)`.

```
{{Infobox officeholder
|honorific-prefix = [[The Honourable]]
|name             = John Howard
|honorific-suffix = [[Member of the Order of Merit|OM]] ...
|image            = Johnhoward.jpg
|office           = [[List of Prime Ministers of Australia|25th]] ...
|monarch          = [[Elizabeth II]]
...
```

Figure 2.4: Excerpt from the Wikipedia article's infobox for `John Howard`.

hyperlinks to other articles, which can viewed as a directed *article graph* that describes how articles—and so entities—are related.

The collaborative process that creates Wikipedia KBs also means that it can be difficult to process; the large number of articles and complex markup mean that "clean" KBs are popular including DBpedia, YAGO and Freebase. While these often draw from Wikipedia content, they may also use other sources or allow direct fact entry. Wikipedia's depth and breadth make it a compelling choice of KB to link against and we do so for this thesis.

Bunescu and Paşca (2006) adopt a search and disambiguation paradigm to automatically link named entity mentions in Wikipedia articles. Detection uses aliases extracted from titles, redirects, disambiguation pages, categories and hyperlinks. Candidates are represented using a VSM model that includes a feature that is the cosine similarity between the candidate's article text and a 55 token window centred around the mention, as well as a taxonomy kernel. This models similarity by generating a feature for the combination of each of the article's categories and terms in the mention's context. A NIL candidate is inserted into the candidate list, with a feature that indicates that it is a NIL.

They use Wikipedia to create a dataset of 1.7M instances where the context around a hyperlink mention is extracted and the hyperlink target taken as a gold-standard referent. They include pseudo-NIL entities where 10% of queries have their correct link artificially omitted from search. They train a Support Vector Machine (SVM) classifier to disambiguate mention candidates, although they note that the combination of features generates extremely memory-intensive models. Cosine similarity is a strong baseline, but the taxonomy kernel accounts for 2.8% improvement to 84.6% accuracy in a setting where NILs must be detected.

As well as evaluating over the Wikipedia gold standard, Cucerzan (2007) explores linking over news articles. The system uses an extraction pipeline that performs case normalisation and NER. The disambiguation takes advantage of mentions from the *whole document*. Context and categories are extracted from Wikipedia articles. Contexts are hyperlink anchors from the first paragraph or reciprocal links (i.e. article A links to article B and vice versa). Categories are taken from a subset of Wikipedia categories and "list" pages. Candidates are then ranked by how well their contexts match the mention's document and their categories match the document vector—categories from all candidates for all mentions.

The combination of textual and KB structured similarities performs well: 88.3% on a sample of Wikipedia articles and 91.4 on news articles from MSNBC. The MSNBC corpus was automatically linked and manually checked, considering only mentions

where the boundary was correctly identified and a KB entry exists, and so the high accuracy may not reflect the performance of an end-to-end system. Cucerzan points out that the ambiguity of Wikipedia link anchor texts is much lower than named entity mentions in news data. This may be because the MediaWiki mark up requires editors to enter the article title in order to make a link, and they must then actively decide to use some other mention string to anchor the text. This seems to encourage them to refer to entities more consistently than writers of other types of text.

These two systems capture some important insights into KB linking. As was the case in CDCR, textual similarity is an important factor, but the structure represented by the KB's categories and article graph provides important evidence for disambiguating entities, and is applicable using supervised and unsupervised methods. Both acknowledge the NIL recognition problem and while only Bunescu and Paşca (2006) account for it, they can be considered seminal NEL systems. We describe and evaluate these systems in Chapter 4.

2.2.2 Wikification

Wikipedia's coverage extends beyond entities and can be considered a collection of concepts, making it suitable for investigating broader concept disambiguation. This has been long studied in the domain of Word Sense Disambiguation (WSD), where words must be grounded to their dictionary entries. This immense body of work[9] informs many of the approaches to linking including context overlap (Lesk, 1986) and linking to structured KBs such as WordNet (Miller, 1995). *Wikification* requires linking mentions of general concepts as well as entities, just as a Wikipedia editor would. Editors may link a mention to its article to add context, but they may only link some concepts or some mentions in the article. An editor may also annotate a mention for which no article exists, which is rendered as a "red link". In contrast to NEL, wikification systems link non-entities and entities, but perhaps not exhaustively and perhaps only non-NIL mentions.

[9] See Navigli (2009) for a comprehensive review.

Mihalcea and Csomai (2007) introduce the wikification task. Keywords are extracted using a number of statistical methods, including "keyphraseness", the conditional probability of a phrase linking to any entity given it appears in an article. Candidates are disambiguated using a voting scheme between a VSM model and naïve Bayes classifier. Their system is evaluated in a Turing test setting where volunteers would attempt to distinguish human and machine wikified Wikipedia pages; they found that they could not. Their use of statistics drawn from the KB which characterise KB link likelihood have been followed in many other systems.

Milne and Witten (2008) learn C4.5 decision tree classifiers that rank candidates based on Wikipedia mention statistics. These take advantage of unambiguous concepts—those with only one candidate—as a context that candidates should match. They use features such as *commonness*, which is the conditional probability of linking to an entity given a specific hyperlink anchor (i.e. $p(entity|mention)$) and average *relatedness* with unambiguous concepts, which takes into account the Wikipedia link graph. They define a general similarity method for Wikipedia articles modelled on Google Normalised Distance (Cilibrasi and Vitanyi, 2007).

$$relatedness(a,b) = \frac{\log\left(max(|A|,|B|)\right) - \log\left(|A \cap B|\right)}{\log\left(|W|\right) - \log\left(min(|A|,|B|)\right)} \tag{2.1}$$

Equation 2.1 shows how relatedness is incorporates inlink overlap, where $|A|$ is the number of hyperlinks that link to article a—its inlinks. While the term $|W|$, the number of articles in Wikipedia, can be pre-calculated, the inlink set calculations requires efficient storage of the article graph. Their system is evaluated in two ways. The bag-of-titles evaluation compares the set of a document's linked mentions with the gold standard, which does not consider where a concept was linked to.

This factors out mention detection, which makes system comparison easy, but may conceal mention detection errors. For example, if ten John Howard mentions in a single document are linked to the wrong KB entry, they are judged the same as one incorrectly-linked mention. They also wikify 50 stories from the AQUAINT

corpus and crowdsource link correctness and relevance judgements using Amazon Mechanical Turk. Just over 75% of links were judged correct, but annotators suggested 8 extra links on average. The *Wikipedia miner* system has been a popular comparison system, as it freely available and successfully combines context evidence to identify and disambiguate concept spans in text.

In order to encourage further research on wikification, the INEX workshops ran a Link the Wiki task in 2007, 2009 and 2010 (Huang et al., 2010). The task is designed to improve Information Retrieval and places an emphasis on Wiki creation and maintenance as well as evaluation tools and methodologies. The 2009 task introduces a second wiki, Te Ara,[10] an expert-edited encyclopaedia about New Zealand. Te Ara does not contain inter-article links, so the first subtask is to discover them. The second task is to link Te Ara articles to Wikipedia articles. A key difference is that systems link to the "best entry point" of the article, which may be a particular section of an article rather than the whole article.

Document-wise evidence improves linking, but the systems above (Cucerzan, 2007; Milne and Witten, 2008) link each mention independently, despite using evidence from the whole document. Kulkarni et al. (2009) propose joint linking models that collectively optimise the links for all mentions in the document, showing the problem to be NP-hard. Their Integer Linear Programming (ILP) and greedy hill climbing approaches build linear SVM models with a range of features: KB statistics, Wikipedia category structure as in Cucerzan (2007) and linking relatedness from Milne and Witten (2008). On a news-domain corpus, IITB, the joint approaches have higher precision at a broader recall range than either of the two compared previous systems, especially the Cucerzan (2007) system, which performs badly.

It is not clear why this is the case; the reported score on the MSNBC data, also news domain, is approximately 25% F[11], low considering is the 91.4% accuracy reported in Cucerzan (2007). Though not directly comparable, one might expect these figures

[10] www.teara.govt.nz
[11] Read from Figure 15

to be closer, and this suggests NER or general replication issues. The joint system performs at 69% F, higher than Milne and Witten (2008) at 63% F. Despite the lower performance gap than on the IITB data, the joint system shows the same attractive recall properties.

The TAGME system (Ferragina and Scaiella, 2010) concentrates on wikifying short texts. This takes advantage of context as in Milne and Witten (2008), but considers all mentions, ranking their candidates by commonness. This is different to Milne and Witten (2008), who only depend on finding unambiguous mentions. On a corpus of short texts generated from Wikipedia, their system improves on Wikipedia Miner by around 3% at 91.2% F, and end-to-end linking around 9% F better. They also find similar accuracy to the system reported in Kulkarni et al. (2009), but at lower complexity as they do not perform joint linking.

Relatedness methods make implicit use of Wikipedia's article graph, but others have made more explicit use, propagating evidence across entity graphs using a personal PageRank (Han et al., 2011), resulting in 73% F, 4% higher than Kulkarni et al. (2009) on the IITB dataset. The AIDA system (Hoffart et al., 2011) use a greedy graph approach to link KB concepts in the CoNLL-03 NER data. This uses the YAGO ontology and syntactic similarity features to jointly link documents. They report better results than reimplementations of Cucerzan (2007) and Kulkarni et al. (2009), and their performance boost is partly attributed to robustness checks that can disable entity popularity features or modify the set of entities used for whole document context. Note that they do not attempt to link NIL mentions. The KORE system (Hoffart et al., 2012) replaces the article graph calculations with keyphrase overlap and uses locality sensitive hashing to avoid calculating semantic relatedness between all candidate pairs, resulting in dramatically improved runtimes at often improved accuracy.

Considering all candidates of all mentions for whole document linking can be computationally expensive, He et al. (2013b) use stacked classifiers: they link mentions locally and use the first classification as features in a globally-aware classifier. Gardner and Xiong (2009) model linking as sequence labelling to explore which spans of text

are hyperlinked in Wikipedia. Their CRF approach is higher precision, but lower recall than Milne and Witten (2008). Cornolti et al. (2013) present a framework for evaluating several publicly available systems on multiple datasets, measuring accuracy, inter-system similarity and efficiency.

Other systems shift the emphasis of whole document linking to earlier in the process: when searching for candidate entities (Pilz and Paaß, 2012). Their system indexes entity names, article text, NE types and article outlinks. All mentions are searched simultaneously and the global result set is post-processed to identify a "best fit" candidate for each mention using relatedness and a coherence score. The candidates are disambiguated using a ranking SVM trained on search rankings, reference probability and LDA topic distributions. Their comprehensive evaluation considers the MSNBC (Cucerzan, 2007), AQUAINT (Milne and Witten, 2008), IITB (Kulkarni et al., 2009), CoNLL-03 (Hoffart et al., 2011) and ACE (Bentivogli et al., 2010) datasets, noting differences that make evaluation more difficult, for example different conventions for deciding the appropriate entity link. They too use a bag-of-titles evaluation, but note that this can obscure incorrect individual links as it aggregates at a document level. They also do not consider NIL links when evaluating over the CoNLL-03 data to match Hoffart et al.. Their systems outperform GLOW (Ratinov et al., 2011)[12], Wikipedia Miner and AIDA, and they observe that, given a coherent set of concepts, collective search is more efficient.

2.3 Entity linking at the Text Analysis Conference

The sections above have introduced the problem of entity ambiguity and several approaches to solve it. Context similarity is important when clustering mentions with one another. When linking to a KB, its structure provides useful information for linking. Evaluating different approaches is difficult as datasets and metrics tend to evolve organically. Some tasks such as WEPS and INEX provide some context for

[12]We discuss this below.

this, but do not explicitly handle NIL mentions. The National Institute of Standards and Technology (NIST) has organised a Knowledge Base Population (KBP) workshop through the Text Analysis Conference (TAC) since 2009 to promote research into entity linking and slot filling, the extraction of specific entity attributes. Entity linking is a query-driven task where systems must identify the entity from a reference KB given a name and document in which it can be found. While this use-case is different from wikification as systems only link one mention per document, it ignores the problem of mention detection and simplifies the evaluation.

This section briefly reviews the first five years of TAC submissions and some work outside the KBP workshop that is evaluated on the same datasets. We discuss the TAC data in Chapter 3 and our systems in Chapter 5. Some systems consult web-based resources, such as the Google Search API during linking. While these often lead to performance boosts, we do not focus on these results as they are difficult to replicate as search result ranking changes over time, is often dependent on location and batch-querying may violate terms of usage. To give insight into how live web-access can help, we report on the performance of the LCC systems from 2010 and 2011, which beat their top-ranked offline systems by between 1% and 2%.

We also do not report on NEL that does not use the KB text, as we see linguistic processing as a key component of linking. Furthermore, much of Wikipedia's information is contained in the article text itself. We also concentrate on monolingual English linking and not the multilingual setting or slot filling, although those tasks are important for real-world KB population.

The teams are drawn from industry: Microsoft Research (MSR), Language Computer Corporation (LCC) and academia: the International Institute of Information Technology in Hyderabad (IIITH), Tsinghua University (THU), Johns Hopkins University (JHU), Stanford University (SU), the University of the Basque Country (UBC), the National Laboratory of Pattern Recognition (NLPR), the National University of Singapore (NUS), City University of New York (CUNY), Heidelberg Institute for Theoretical Studies (HITS), University of Illinois at Urbana-Champaign (UIUC), Macquarie

University (MQ) and University of Sydney (USYD).[13] Many teams compete in multiple years, building on their previous work and the engineering investment required to produce even a baseline system.

2.3.1 2009

The first shared evaluation (McNamee et al., 2009b) took place in 2009. Teams were provided a reference KB derived from a Wikipedia snapshot from October 2008. The TAC KB contains 818,741 entries created from articles with infobox markup. An entry consists of a name, an automatically assigned entity type, an entity ID, a list of slot name-value pairs extracted from infoboxes and the article text with markup expanded. The mention documents are from newswire and web sources around the same time as the KB snapshot was taken. The evaluation dataset consists of 3,904 queries consisting of a mention string and the document in which it can be found.[14] Systems return the appropriate entity ID or NIL and are evaluated using micro-averaged accuracy.[15] This considers a KB query correct only if the correct ID is returned, but a query linking to an entity outside the KB only requires the system to return NIL rather than cluster mentions as is the case in later years.

Some teams adopt a *Wikipedia mapping* approach to TAC, first linking to a larger Wikipedia snapshot, before mapping back to the TAC KB. More recent snapshots of Wikipedia are larger and richer than the derived KB, but this approach has some implications for task realism, which we discuss in Chapter 5.

Teams were provided the KB and corpus data for 4 weeks followed by a 2 week evaluation period. Table 2.2 shows the top 5 results by accuracy[16]. Systems mostly use the mention name from the query, with some using NER, coreference resolution (Bikel et al., 2009) and acronym detection (Varma et al., 2009; Li et al., 2009) to find other mentions of the query name in the document. Candidates are searched over

[13] This list is not exhaustive.
[14] See Section 3.2 for more details and examples of the datasets.
[15] Macro-averaging over all queries for an entity was also reported in TAC 09.
[16] As we are reviewing competitions, we only show the top non-web system for each team.

System	Accuracy		
	All	KB	NIL
Varma et al. (2009, IIITH)	82.2	76.5	86.4
Li et al. (2009, THU)	80.3	77.3	82.6
McNamee et al. (2009a, JHU)	79.8	70.6	86.8
Agirre et al. (2009, SU,UBC)	78.8	75.9	81.1
Han and Zhao (2009b, NLPR)	76.7	69.3	82.3

Table 2.2: Results from TAC 09.

Wikipedia titles, redirects, disambiguation pages and hyperlink anchors, often using a full-text index, with some using a complex arrangement of search components (Varma et al., 2009; Honnibal and Dale, 2009).

Many systems use a VSM model as their main disambiguating feature using the mention's document tokens and candidate article text (Varma et al., 2009; Honnibal and Dale, 2009; Han and Zhao, 2009b), but some systems experiment with supervised linking despite the lack of training data provided in the task. Classifiers include SVMs with linear (Agirre et al., 2009) and polynomial kernels (McNamee et al., 2009a) and the list-wise ranker ListNet (Li et al., 2009). Li et al. (2009) also learn a separate SVM classifier that predicts whether the top disambiguated candidate is a NIL or a KB query, and later report a score of 85.0% accuracy on the TAC 09 dataset Zheng et al. (2010). The HLTCOE team present a thorough description of their supervised system (McNamee et al., 2009a; Dredze et al., 2010). They compiled their own dataset for training, and learn SVM classifiers over a large collection of features, including popularity in a Google Search API call, related entities extracted using the SERIF information extraction system (Boschee et al., 2005) and Wikipedia statistics such as page size, number of inlinks and outlinks. They find that popularity features and KB statistics help linking. They also evaluate over the Cucerzan (2007) MSNBC data yielding 94.7% accuracy, higher than Cucerzan's 91.4% accuracy. Agirre et al. (2009) heuristically-weight the output of a dictionary based approach, a linear SVM, VSM-

derived cosine similarity and Google Search API score. Their SVM model is trained on Wikipedia, harnessing editor annotations as a gold standard. Han and Zhao (2009b) manually combine simpler measures: a VSM cosine similarity and a semantic similarity approach based on Milne and Witten (2008).

Fisher et al. (2009) use a CDCR approach to cluster query documents, Wikipedia articles and other matching documents from the source corpus using TF-IDF weighted terms. To find a query's entity ID, the Wikipedia article closest to the document's cluster is found. If this corresponds to a TAC KB entry, that entity ID is returned. Their method performs reasonably well at 65.9% and is notable for adopting a purely clustering approach and taking advantage of unlabelled text.

The organisers note several tricky query types (McNamee et al., 2009b). Subsidiary organisations can be hard to identify, as their names may be composed of an ORG followed by a specialising LOC (e.g. Virgin Australia), which may be recognised as two separate entities. While people are discrete entities, organisational relationships can be intricate, or a parent company may be in the KB and the subsidiary a NIL. Typographical mistakes, acronyms and metaphoric names (e.g. The Iron Lady) can lead to recall problems where the correct candidate is not retrieved in the search phase. Metonymy remains challenging, especially when city names are used to refer to their sports teams. Finally, manual NEL annotation is challenging, and involves marking the mention span and type (as with NER), but also searching the TAC KB for candidate matches. Some disambiguation decisions can be subtle and it is no surprise that the gold standard contains some errors.

2.3.2 2010

The 2010 task requires systems to link over web documents as well as newswire and the optional "no text" version of the task, where systems are prohibited from using the text field of the KB entry. Table 2.3 shows the top 5 performing submissions.

System	Accuracy		
	All	KB	NIL
Lehmann et al. (2010, LCC)	85.8	79.2	91.2
Radford et al. (2010, USYD)	81.9	73.7	88.7
Varma et al. (2010, IIITH)	81.7	71.6	90.2
McNamee (2010, JHU)	81.4	75.3	86.3
Chang et al. (2010, SU,UBC)	80.0	65.1	92.4

Table 2.3: Results from TAC 10 without web access.

Despite the availability of training data, 4 of the top 5 systems that do not use the web were unsupervised. Lehmann et al. (2010) use the regular suite of search sources from Wikipedia (i.e. titles, redirects, hyperlink anchors and disambiguation pages), as well as a Google Search API call. They use a simple heuristic ranking followed by supervised disambiguation using relatedness features (Milne and Witten, 2008), checks in DBpedia, the genre of the mention document (web or newswire) and the source of the search match (a high precision source like title, etc.). Their live web-access fully supervised system scores 86.8% and 86.4% accuracy varying a prior on NIL queries. LCC submitted the only live web-access system to beat an offline system in TAC 10, but their simpler heuristic system performs surprisingly well at 85.8%. Our system (Radford et al., 2010) is also unsupervised and shows similar performance to Varma et al. (2010), who use cosine similarity disambiguation.[17] We report our system in more depth in Chapter 5. McNamee (2010) re-engineer their TAC 09 system for high recall and to use fewer features, removing Google Search API features. They include acronym matching (Li et al., 2009) and use an interesting NIL feature that specifies if no candidate has any co-occurring NEs with the mention document. Chang et al. (2010) extend their large-dictionary search approach, experimenting with fuzzier matching and deterministic coreference. Rather than training one disambiguation

[17] Their web-allowed system scores 83.7%.

model for the system, they train a separate model for each ambiguous query string. Their best system uses exact mention search and scores 80% accuracy.

Other systems investigated entity attributes from slot values (Chen et al., 2010) and matches in infobox fields (Goldschen et al., 2010). Graph metrics were also used to capture higher quality matches against the KB (Goldschen et al., 2010; Fernández et al., 2010). Yu et al. (2010) adopt a margin-based model for NIL classification, only returning an entity ID if there is a significant margin between it and the second or third candidates. Overall, as in TAC 09, systems perform well, and Ji and Grishman (2011) provide an overview of the task. They note that ambiguous geopolitical entities are problematic and they propose that following hyperlinks from the mention document is an under-utilised source of disambiguating context.

2.3.3 2011

The NEL task changed substantially in TAC 11, adding NIL clustering and multilingual linking. Rather than being simply identified, NIL queries should be clustered, which is essentially a CDCR task that can include KB entries. This task change requires a new metric—B^{3+}—adopted from coreference resolution research (Bagga and Baldwin, 1998a) and explained further in Section 3.4. As with the original B^3 metric, each query is scored on how well its cluster neighbours agree with the gold standard, with a restriction to ensure that KB queries link to the correct entry[18]. Table 2.4 shows the top 5 systems by B^{3+} F, with micro-averaged accuracy where it is available. Each of the top systems is supervised—teams had access to training data from TAC 09 and TAC 10 (training and evaluation). There is a bewildering array of learning approaches used in TAC systems and it is difficult to compare them directly as they only form one part of often complicated pipelines. Although their results are outside the top 5 at 71.2%, Anastácio et al. (2011) present a thorough comparison of different learning methods, finding no clear winner across different datasets.

[18]Otherwise we could have perfect clustering of KB queries, but link them to the wrong entries.

System	Acc.	B^{3+} F		
	All	All	KB	NIL
Monahan et al. (2011, LCC)	86.1	84.6	76.2	93.0
Cucerzan (2011, MSR)	86.8	84.1	78.3	89.9
Zhang et al. (2011b, NUS)	86.3	83.1	75.3	90.9
Cassidy et al. (2011, CUNY,UIUC)	-	77.1	64.1	90.0
Ratinov and Roth (2011, UIUC)	78.8	76.1	61.1	91.0

Table 2.4: Results from TAC 11 without web access. Accuracy is not always reported.

Monahan et al. (2011) use their 2010 linker (Lehmann et al., 2010), which links heuristically with supervised models for ranking and NIL classification. They explore two clustering methods: inductive, which considers all linked queries, and deductive, which clusters only NIL mentions. Their best system uses an inductive approach, which is able to repair bad linking decisions by finding similar contexts. The clustering is multi-stage, first by name, then supervised hierarchical agglomerative clustering as per Culotta et al. (2007). The supervised clustering features include entity type, linked entity ID, cosine similarity. Finally, clusters are merged depending on whether they link to the same article or are contained in the same noun phrase, scoring the best B^{3+} F at 84.6%. As in 2010, they also submitted systems that used live web-access and these again beat their offline system with B^{3+} F of 86.9% and 86.4%.

The second-ranked system by B^{3+} F, but first by accuracy, is based on an extension of the system presented in Cucerzan (2007). The system takes a whole document approach, linking to a larger KB. Topics in the extended system include lexicosyntactic patterns and tokens from category names instead of contexts to avoid traversing the article graph. They use a linear combination of Wikipedia prior, similarity between context and topic, similarity between candidate topics and the aggregated document topics, number of surface forms, whether parenthesised content is included. The model is trained on the TAC 10 training data, and scores 90% accuracy on the TAC 10 evaluation data. Rather than sophisticated clustering, they rely on accurate linking to

a larger Wikipedia, then clustering to the article title. If this is not in the TAC KB, then a NIL ID is generated for all queries that link to it. Where no link is made, the queries are clustered by mention.

Zhang et al. (2011b) expand acronyms to extract better mentions for search. Disambiguation uses a ranking SVM and supervised NIL classifier, using features such as name matching, context similarity, NE similarity between mention document and candidate article, topic similarity. They build a Wikipedia dataset using iterative batch selection and combine spectral, HAC and LDA clustering scores using an SVM.

The remaining two prominent systems both use GLOW (Ratinov et al., 2011), a system that links using local and whole document features. The authors situate the work between Cucerzan (2007), where the document's disambiguation context contains all candidates including many erroneous ones and Milne and Witten (2008), which depends on finding sufficient unambiguous mentions. Their two stage process uses a ranked SVM to link mentions using local features such as text similarity, and a re-weighted score conditioned on the set of candidates for a mention. Global features are calculated across the article graph of the linked mentions for final disambiguation. They evaluate the systems on the AQUAINT dataset from Milne and Witten (2008), scoring up to 95.6%. Their best approach on the Cucerzan (2007) MSNBC dataset scores 88.5%, lower than reported in Cucerzan due to lower-recall search. They also evaluate using a bag-of-titles metric, where a set-based F is calculated with respect to the gold standard to factor out differences in mention boundaries. GLOW performs better than Wikipedia Miner, but global linking does not improve the strong baseline of local approaches by a large margin. Ratinov and Roth (2011) frame linking as a post-process over GLOW output, assigning NIL to improve the value of an objective function. Cassidy et al. (2011) vote over the output of the CUNY system, a combination of supervised rankers, and GLOW responses. These are combined using a collaborative clustering framework (Chen and Ji, 2011) where extra context is sought from other queries for clustering.

System	Acc.	B^{3+} F		
	All	All	KB	NIL
Cucerzan (2012, MSR)	76.6	73.0	68.5	78.1
McNamee et al. (2012, JHU)	-	69.9	65.3	74.9
Tamang et al. (2012, CUNY)	-	68.8	59.5	78.9
Monahan and Carpenter (2012, LCC)	75.7	68.5	59.2	78.7
Radford et al. (2012, USYD)	-	66.5	65.6	67.5

Table 2.5: Results from TAC 12 without web access. Accuracy is not always reported.

Many systems use entity information: from a 10-token window around the mention (Fahrni et al., 2011) extracted slot values (Cassidy et al., 2011), or fine-grained semantic type annotation with a DBpedia tool (Mendes et al., 2011). Using entity-mediated context was popular, either from the whole document (He and Wang, 2011; Ratinov and Roth, 2011) or isolating the location (Cao et al., 2011; Zhao et al., 2011) or time (Cao et al., 2011) of a document. The TAC 11 competition saw a real emphasis on supervised systems and arrays of features modelling a mention's context and KB structure. The NIL clustering task attracted a range of approaches from heavyweight clustering to simple rule-based systems.

2.3.4 2012

The 2012 TAC task makes only minor changes to the NEL specification, providing offsets for the mention. This supports annotation of more ambiguous queries, where the annotators were not restricted to entity mentions and could use "any textual extent" (Ellis et al., 2012). Another variant is the "cold start" task, where systems must populate a KB from scratch. Table 2.5 shows the top 5 results by B^{3+} F.

The top system in 2012 (Cucerzan, 2012) is an extension of the second ranked system in 2011 (Cucerzan, 2011). This system delays fixing mention boundaries until late in the disambiguation phase so as to recover from NER errors, and also adds richer context features for geolocation and concepts. McNamee et al. (2012) present the

Context Aware Linker of Entities (CALE), a joint method that uses structured prediction cascades to minimise the number of mention candidates and constructs a markov random field over mentions in the same paragraph to jointly link them. Tamang et al. (2012) extend their collaborative ranking system (Artiles et al., 2011), which uses an ensemble of naïve rankers, adding query reformulation and collaborative clustering. This incrementally adds new instances to improve clustering quality, accounting for limited context that can make clustering mentions difficult. They also find *V-measure* (Rosenberg and Hirschberg, 2007) better suited for cluster evaluation than B^{3+}.[19] LCC concentrates on cold-start KBP with the Lorify system (Monahan and Carpenter, 2012). This uses their existing linker (Lehmann et al., 2010; Monahan et al., 2011) and they experiment with large-scale clustering approaches: HAC and Markov Chain Monte Carlo sampling, their best system using the former.

Systems introduce precise disambiguation features based on descriptions of an entity, those extracted using REVERB, an open IE system, (Bonnefoy and Bellot, 2012), a candidate's "appositional" terms in the text such as `Illinois` in `Toronto, Illinois` (Graus et al., 2012; Clarke et al., 2012), or other mentions close to the mention and matches from the candidate's infobox (Clarke et al., 2012). Clarke et al. (2012) also attempt to directly estimate how ambiguous a mention is using an unsupervised model that performs coreference resolution on a corpus and calculates relatedness with respect to the corpus rather than the KB. They frame linking as online clustering seeded with KB entries, and use structured prediction to predict whether to merge a mention into a cluster. Fahrni et al. (2012) compare a 15-token window around a mention with the same window extracted for all inlink anchors for a candidate. Where a mention occurs in multiple documents, they extract a context of noun and adjective n-grams from each mention's containing phrase. Any overlap between the local syntactic context indicates mention similarity. These and other features are used to within Markov Logic Network, which jointly links and clusters mentions, reporting 71.8% accuracy and 62.1 B^{3+} F. They also report results on other datasets (Fahrni

[19] We do not present comparison here as it is not an official TAC metric.

and Strube, 2012), linking the TAC 11 data at 82.9% accuracy and 80.1 B^{3+} F, and the ACE 2005 (Bentivogli et al., 2010) with 74.3% accuracy, where an upper bound is 93% based on the candidates returned from search.

Although many systems query a full-text index to retrieve candidates, IR techniques are usually not the focus of a system. Against this trend, Dietz and Dalton (2012) use cross-document context and probabilistic IR to retrieve small, high-recall candidate sets. The mention context is used to retrieve similar sentences in the corpus and these augment the query. Mentions are disambiguated using a learning-to-rank framework over features that capture the similarity of a mention and candidate name, context and KB statistics. Although they submitted a non-optimal system, they report post-competition accuracy of 71.3%. This follows from earlier work using a statistical language model to disambiguate entities (Gottipati and Jiang, 2011), which scores 85.2% on the TAC 10 dataset.

The TAC 12 datasets were more difficult by design and scores are correspondingly lower than TAC 11 by roughly 10 B^{3+} F. While the best system is solely oriented towards accurate linking, sophisticated clustering and joint models perform competitively.

2.3.5 2013

Coverage of the 2013 TAC task is brief as it was completed before the results and proceedings were available. The 2013 task includes documents from discussion forums and systems are permitted to include a confidence value for up to five links per query. Table 2.6 shows the top 5 results by B^{3+} F. Cucerzan and Sil (2013) add features to their TAC 12 system that model the local context of entity mentions and are able to induce an entity type distribution over words. Their top score of 74.6% B^{3+} F is the highest of all systems, and they also report impressive accuracies on TAC 11 (89.9%) and TAC 12 (79.3%). The team from the University of Sydney, Pink et al. (2013), report a version of the supervised system presented in this thesis that uses a simplified version of the local features that we describe in Chapter 7, but with a separately learned model

System	Acc.	B^{3+} F		
	All	All	KB	NIL
Cucerzan and Sil (2013, MSR)	83.3	74.6	72.2	77.2
Pink et al. (2013, USYD)	83.1	72.7	71.4	73.8
Wang et al. (2013, THU)	-	71.2	72.1	70.0
Cheng et al. (2013, UIUC)	-	69.4	68.6	70.0
Fahrni et al. (2013, HITS)	81.7	68.4	67.8	68.1

Table 2.6: Results from TAC 13 without web access. Accuracy is not always reported.

for queries labelled with PER than for those tagged with other labels. The best score is 72.7% B^{3+} F, interestingly without the early version of the local features. The system submitted by Wang et al. (2013) use a supervised listwise ranking model and a collaborative ranking approach to cluster entities, scoring 72.1% B^{3+} F. Cheng et al. (2013) combine an NEL and dedicated CDCR system, which performs well at 69.4% B^{3+} F, while Fahrni et al. (2013) extend their Markov Logic Network system to link common nouns, which obtains a B^{3+} F of 68.4%. TAC 13 sees a diverse range of systems in the top 5 systems, that so many are extensions of systems from previous years is an indication of the maturity of the approaches.

2.3.6 Beyond TAC

The shared task environment has prompted a diverse range of NEL approaches, but is not the only venue for work that evaluates using TAC queries. We have discussed some work from outside the competition above and summarise other major work below. These fall into four broad categories: using the KB graph, abstracting from lexical to topical context, training data creation and characterising a mention's disambiguating content. Table 2.7 shows the best (non-web) performance in different years of TAC and the results of other systems that have since evaluated on the same datasets.

Wikipedia's link structure, in particular, has driven new approaches incorporating graph-based methods for NEL. This is the motivation behind citation overlap

System	Data	Accuracy	B^{3+} F
Varma et al. (2009) (TAC)	09	82.2	
Shen et al. (2012a)	09	78.5	
Zhang et al. (2010)	09	83.8	
Ploch et al. (2011)	09	84.2	
Shen et al. (2012b)	09	84.3	
Zheng et al. (2010)	09	85.0	
Han and Sun (2012)	09	85.4	
Han and Sun (2011)	09	**86.0**	
Lehmann et al. (2010) (TAC)	10	85.8	
Guo et al. (2011)	10	82.4	
Zhang et al. (2012)	10	87.8	
Cucerzan (2011)	10	**90.0**	
Cucerzan (2011) (TAC)	11	86.8	-
Monahan et al. (2011) (TAC)	11	-	84.6
Fahrni and Strube (2012)	11	82.9	80.1
Cheng and Roth (2013)	11	86.1	83.7
Zhang et al. (2012)	11	**87.6**	-

Table 2.7: Results outside TAC competition over different TAC datasets. We show the best TAC systems by accuracy and B^{3+} F where reported for TAC 11.

measures between candidates and unambiguous context entities (Milne and Witten, 2008; Lehmann et al., 2010; Ratinov et al., 2011). More recent systems build a graph where vertices correspond to mentions and/or their entities and edges correspond to candidate entities for given mentions and/or entity-entity links from Wikipedia. Intuitively, highly connected regions represent the "topic" of a document and correct candidates should lie within these regions. Ploch (2011) demonstrate that PageRank (Brin and Page, 1998) values for candidate entities are a useful feature in their supervised ranking and NIL detection systems, leading to an overall accuracy of 84.2% on the TAC 09 data. Hachey et al. (2011) show that degree centrality is better than PageRank, leading to performance of 85.5% on the TAC 10 test data. And Guo

et al. (2011) show that degree centrality is better than a baseline similar to the cosine baselines reported in Chapter 4, leading to performance of 82.4% on the TAC 2010 test data. Shen et al. (2012a) link concepts in a window around the mention and use these, the mentions and candidates as nodes in a graph. Each candidate is given a label which is propagated back to the mention to link them, scoring 78.5% in TAC 09. Their system, LINDEN (Shen et al., 2012b), depends heavily on the concept graph to link mentions in lists, a limited context, and scores 84.3% on TAC 09. Graph-based approaches are popular and powerful, however they require substantial processing at linking time to query and traverse the graph. Moreover, they depend on a rich graph structure which may not be present in all contexts, for example when linking to a different KB, or during population, where deciding how to connect a newly added NIL entity to the rest of the graph is an open research question.

Mention context is an important signal for disambiguation, but often lexical similarity can suffer from sparseness. Generalising the context terms to higher level topics can help solve these problems and has been used in CDCR, wikification and in TAC competition. Zhang et al. (2011c) learn a topic model that proposes a distribution of Wikipedia categories for a mention's context words. The distribution is combined with a more refined method for generating training data from Wikipedia akin to active learning and scores 87.6% on the TAC 10 dataset. This is similar to a generative model of linking (Han and Sun, 2011) where a linked entity is the result of an entity name and context generated from an initial KB entry, scoring 86% accuracy evaluated on TAC 09. Han and Sun (2012) extend this to jointly model topic and context using a Gibbs sampling approach. This approach scores 80% F on the IITB dataset and 85.4% on TAC 09, but does not address NIL detection or report on other TAC datasets. He et al. (2013a) use deep neural networks to learn a similarity between a document and an entity that embeds semantics. They score 81.0% accuracy on TAC 10 and exceed the performance of several collective approaches on the CoNLL dataset.

Early supervised approaches were hampered by the lack of training data, and Wikipedia has been used as a substitute. This can be challenging and lead to data

that does not match TAC. Wikipedia's encyclopaedic style means that mentions and their contexts may not reflect the news and web data used in TAC. The distributions of KB and NIL queries may not match test TAC queries, it is rare to take advantage of "red links" that refer to an article that does not exist. Zhang et al. (2010) combine TAC and Wikipedia training data to good effect. Based on a high-recall search, including using the Google Search API, they create training data from the TAC corpus by finding unambiguous, well-specified mentions, then replacing them with ambiguous equivalents (e.g. John Howard might be replaced with Howard), generating positive and negative instances. These are added to instances drawn from Wikipedia articles, disambiguated by hyperlinks. Disambiguation uses an SVM model with bag-of-words features, Bunescu and Paşca (2006) term-category pairs and NE type match features. This scores 83.8% accuracy on the TAC 09 data and they propose that this is a result of training the system on a large dataset that is representative of the test data. They further experiment with deferring learning until disambiguation with a "lazy learning" technique that generates query specific training data from a mention's candidates (Zhang et al., 2012). This shows a 3.8% improvement on the TAC 10 data for a final score of 87.8% and 87.6% on TAC 11.

Entities are typically described, especially when introduced in a discourse, to provide disambiguating context and general information. Harnessing such descriptions can help disambiguate mentions. Cheng and Roth (2013) extract related entities as descriptions for NEL. After identifying mentions and candidates, they solve an Integer Linear Program (ILP) that jointly optimises linking and relation assignments. This uses syntactic, coreference and relational features, including whether mention apposition can be found in the candidate article. For example, given a mention John Howard, the former Prime Minister,..., a bag of words of the phrase the former Prime Minister would be compared against the candidate article bag of words. They compare this system against Wikipedia Miner (Milne and Witten, 2008) and GLOW (Ratinov et al., 2011) on the MSNBC and AQUAINT corpora. Jointly extracting relations helps performance in TAC 11, their system scoring 86.1% accuracy and 83.7% B^3 F, competitive given it is

not trained on the TAC corpus or Wikipedia. Li et al. (2013) attempt to learn disambiguating description, proposing a generative model where disambiguating terms are generated by latent topics. They train on hyperlinks between Wikipedia pages and from the web. They evaluate over a subset of TAC 09 KB queries and a corpus of 340 tweets, outperforming AIDA (Hoffart et al., 2011), GLOW (Ratinov et al., 2011) and TAGME (Ferragina and Scaiella, 2010).[20] Infoboxes contain some disambiguating facts, but these may only be the subset of those specified in the text. Garera and Yarowsky (2009) use a variety of approaches to extract biographic data from Wikipedia articles. Bootstrapped patterns (e.g. [X] was born in [Y]), positional information (birth dates are usually specified before death dates) and terms that correlate with infobox fields are used to extract facts from the articles. Wikipedia's structure is also used to capture transitive facts, as entities mentioned together often share similar attributes, and identify facts that are unlikely to co-occur. Textual entity descriptions are important cues for disambiguation and can also be useful in the broader KB population context, where characterising a newly identified NIL entity is important.

2.4 Beyond linking text to Wikipedia

The five years of TAC that we have described above have prompted a diversity of approaches to the entity linking problem. A large factor in the KBP workshop's success is its framing of the task as a query-based linking and common evaluation. We have focused on linking source text to Wikipedia, but in this section, we explore other variations. These include linking structured sources, linking streaming data, linking to index and using different target KBs.

2.4.1 Linking structured sources

The tasks above assume textual context for an entity mention, but this is not always the case. Record Linkage, surveyed in Winkler (2006), aims to merge entries (e.g.

[20] They report problems integrating the mentions produced by TAGME for evaluation.

with names and addresses) within one database or across multiple databases. This is often framed as database cleaning: canonical versions of names and addresses are produced, with duplicates sometimes removed in the process. Initial research by Fellegi and Sunter (1969) presented a probabilistic description of the linkage problem and subsequent work extends this to use multiple sources of information or treats it as a graph of mentions to be partitioned into entity clusters. Bhattacharya and Getoor (2007) use a collective, or joint, approach to relational entity resolution. They use this to simultaneously cluster entities in a citation graph. This does, however, allow exploration of large datasets of person-related data (e.g. census and medical records), motivating work on efficiency.

Open Information Extraction systems aim to extract tuples from large-scale corpora and rely on redundancy to identify salient relations between entities. Lin et al. (2012a) link entities mentioned in extracted REVERB tuples to Wikipedia. This abstracts over single documents by aggregating context from sentences that a tuple is extracted from. Web-scale processing requires a lightweight linking model and the authors report linking at around 60 tuples per second at over 70% accuracy using context similarity, KB statistics and statistics gathered from linking large corpora.

2.4.2 Linking streaming data

Davis et al. (2012) disambiguate entities in streaming data (tweets) using an incremental classifier; other systems address the extremely noisy mention detection problem in twitter (Guo et al., 2013a). Liu et al. (2013) use context from other tweets to help disambiguate pre-extracted mentions to KB entities in a corpus of around 500 tweets. They find that inter-tweet context is useful, but this work rests on the assumption that NER has already been performed and similar tweets have been identified; both tasks are challenging at realistic levels of data noise and scale. Guo et al. (2013b) also use inter-tweet context, propagating entity link labels between tweets. Their task is framed as query-based and uses retrieval of relevant tweets to filter noisy tweets.

They find that the sparser context in tweets makes linking more difficult than news, but content expansion (similarity between query results is assumed more tractable than across the whole stream) and propagation techniques improve a VSM baseline.

2.4.3 Linking to index

Linking is popular in the biomedical domain, where it helps to index a vast body of literature. Aronson (2001) describe MetaMap, a system that links mentions to the Metathesaurus, a large biomedical thesaurus. The 2008 BioCreative workshop ran an entity linking challenge for biomedical text, which they termed Gene Normalisation (GN; Hirschman et al., 2005; Morgan et al., 2008). Participants were provided the raw text of abstracts from scientific papers, and asked to extract the Entrez Gene identifiers for all human genes and proteins mentioned in the abstract. The GN task is motivated by genomics database curation, where scientific articles are linked to the genes/proteins of interest. The GN task differs from the real curation task in that it does not use the full text of the articles, and it annotates every human gene/protein mentioned (not just those described with new scientific results).

Other tasks consider linking without mentions, instead assigning KB entities as "tags" for a document. Bhattacharya et al. (2008) learn a joint model that identifies the film title from its review, linking the whole document rather than a specific mention. In the TREC Entity Ranking task, systems must retrieve the URL of a query entity's primary web page. Kaptein et al. (2010) first link to the query's Wikipedia article, then follow hyperlinks to find the primary page that is the final response. Blanco and Zaragoza (2010) discuss information retrieval approaches to finding entity "support sentences", which explain the relationship between an entity and an ad hoc query. For example, the entity `Franco` and query Spanish Civil War might yield the support sentence In 1936, Franco participated in a coup d'état against the elected Popular Front government. Their approach uses BM25 ranking, but they nominate that more sophisticated notions of context are desirable.

Finally, the relation between mention and referent may not be identity, but relevance. Zaragoza et al. (2007) rank entities returned for queries into most important, important and related categories. Nothman et al. (2012) introduce the idea of event reference as linking and discusses several problems. Perhaps the most significant is identity; where annotators may largely agree on what constitutes a particular entity, the same cannot be said for events. Events can contain other events and have complex logical relations to other events.

2.4.4 Linking to other KBs

We have concentrated on systems that link to Wikipedia, but these are not the only KBs suitable for linking. Han and Zhao (2010) take a general approach to calculating concept similarity over a graph of concepts built from mentions in the WEPS data, Wikipedia and WordNet, dubbed "Structural Semantic Relatedness" (Han and Zhao, 2010). This uses the inlink-overlap measure from Milne and Witten (2008) and Cilibrasi and Vitanyi (2007), and continues earlier work where they found the Wikipedia article graph to improve linking over bag of words and social network approaches (Han and Zhao, 2009a). Sil et al. (2012) investigate linking text to arbitrary knowledge bases using distant supervision and domain adaptation. Toponym resolution (Leidner, 2004) aims to link location mentions to a KB entry that represents its exact coordinates, which shares some features of Wikipedia-based NEL simply as Wikipedia has good coverage of locations generated from gazetteers. Moreover, famous locations typically have large, detailed articles.

Freebase is, circa 2012, around 5 times larger than Wikipedia and Zheng et al. (2012) learn a discriminative Maximum Entropy model over Wikipedia sentences where a hyperlink anchor to a Wikipedia article is assumed to link to its Freebase equivalent at 90% accuracy in a corpus of Wikipedia sentences. This takes advantage of unambiguous mentions and Freebase's rich taxonomy, as well as its size. The larger KB has encouraged systems that attempt to link "long tail" and emerging

entities. ClueWeb[21] is a large web corpus and several systems link its documents to Freebase (Mohapatra et al., 2013; Gabrilovich et al., 2013). Entity-centric KBs may not be available to link against, and Zwicklbauer et al. (2013) link against a document-centric KB composed of annotated documents and report good results given a sufficiently large corpus. This is similar to CDCR, and assumes a KB that is, in a sense, latent and the documents that link to an entity represent its entry.

We have concentrated on linking mentions to Wikipedia, but, as shown above, this is only one configuration of the linking task. The possible settings are really only limited by the texts and KBs available.

2.5 Applications

While accurate entity disambiguation is an interesting research goal, it can improve performance in other NLP tasks and be used directly in applications.

Although named entity recognition and coreference resolution usually benefit linking, the reverse can also be true. Linking can benefit entity recognition, where link candidates help resolve entity type ambiguity (Stern et al., 2012). Clustering nominal mentions is a major challenge for coreference resolution and systems have attempted to link to external KBs that might contain nominal aliases for an entity. Systems have linked into the YAGO ontology (Rahman and Ng, 2011; Uryupina et al., 2011) and Wikipedia (Ratinov and Roth, 2012) to incorporate external knowledge. To account for linking errors, Zheng et al. (2013) aggregate votes from chain mentions to help clustering. Hajishirzi et al. (2013) integrate linking constraints into the Stanford sieve system (Raghunathan et al., 2010), using the output of GLOW and Wikipedia Miner. The constraints ensure that mentions that link to the same KB entry are clustered, and they used fine-grained attributes from Freebase to help merge nominal mentions with linked name mentions. Linking improves performance on standard coreference resolution datasets above the basic sieve system by 0.3% B^3 using automatically-

[21] http://lemurproject.org/clueweb12

extracted mentions on the CoNLL 2011 test data. They also show that the lack of coreference resolution in GLOW and Wikipedia Miner hampers NEL performance. Nastase et al. (2012) use Wikipedia concept networks to help resolve cases of name metonymy, themselves difficult cases for linking, as a country name can stand for the national sports team.

As entities and general concepts are important in many text collections, resolving ambiguity allows documents to be indexed by the entities that they contain. This information can drive timelines (Mazeika et al., 2011), and co-occurring entities form graphs to explore entity relations (Malik et al., 2011; Hossain et al., 2012). As well as the news domain, wikification can help browse scientific (Lioma et al., 2011) and cultural heritage corpora (Fernando and Stevenson, 2012). Taneva et al. (2011) retrieve images of long-tail entities with ambiguous names by identifying disambiguating phrases from their Wikipedia articles with which to refine queries.

Errors are inevitable in any linking system and so applications may need human intervention to check and correct links. Fernández et al. (2007) describe a media use-case where journalists need to add entity metadata to their stories for automatic indexing and promotion. Entities are automatically disambiguated, manually corrected and kept for training purposes. Their system takes advantage of entity context, categorical metadata and temporal information to disambiguate entities using a personalised PageRank. Other systems incorporate corrections (Wick et al., 2012; Wang et al., 2012) and crowdsourcing (Demartini et al., 2012) to help improve linking decisions.

Finally, applications must have strategies for handling mentions that are not easily linked. Lin et al. (2012b) identify unlinkable noun phrases—those that cannot be linked to Wikipedia. These are classified as entities or non-entities and their semantic type identified. Emerging NIL entities are another important class of mentions, because they are likely candidates for new KB entries. Nakashole et al. (2013) use bootstrapped patterns to label emerging NILs with fine-grained semantic types. These and other work merging distributional semantics with KBs using linking (Gardner, 2012) hint at solutions for automatic knowledge base curation.

2.6 Summary

This section reviews how the problem of name ambiguity has been explored in several streams of research. Mentions have been compared with one another in Coreference Resolution, both in-document and across a corpus of documents (CDCR). Matching mention context is a key element to this work and CDCR has prompted efficient solutions. Introducing a KB as a pivot allows disambiguation to use context as rich as the KB. Linking entity and general concept mentions to Wikipedia has taken advantage of the article graph and statistics derived from it, as well as the text itself. The TAC shared task has driven progress in systems and refined its task definition over time. We also note a variety of entity-oriented applications benefitting from accurate disambiguation. Through this investigation, we have seen that context is critical for resolving ambiguity and systems are moving towards extracting and disambiguating on the basis of increasingly precise entity attributes. We believe that the manner and content of entity description is an important factor for improving entity disambiguation and explore this further in Chapters 6 and 7.

3 Evaluating entity linking

```
Tony Smith (rugby league born 1967)
Tony Smith (rugby league born 1970)
Brian Smith (rugby league)
Brian Smith (rugby union)
```

Ambiguous Smiths in Wikipedia.

This chapter outlines how entity linking is evaluated: what datasets we measure performance against and how we measure it. The three tasks, CDCR, wikification and NEL, make different assumptions about the problem of name ambiguity, leading to different datasets and metrics. Table 3.1 gives an overview of datasets used in related areas. These vary in text type, whether they annotate the whole document and the size of the dataset (number of mentions). CDCR groups mentions by their context and uses clustering metrics, such as B^3 (Bagga and Baldwin, 1998a). Wikification considers all mentions in the document, but since it links to a KB (i.e. Wikipedia), systems can measure the set of KB entities retrieved. The TAC task frames linking as a query-driven task, specifying just one mention per document. As well as linking to the KB, it addresses the problem of NILs: identification and clustering, which is more strongly linked to CDCR.

This chapter introduces the CDCR datasets, which do not link against a KB. We then focus on the TAC task, describing their query-based datasets, followed by some other Wikipedia-oriented datasets. We then define some metrics that are used to evaluate systems against the gold-standard datasets. We conclude with some observations on

Task	Name	Year	Genre	Whole doc.	KB	Mentions
CDCR	John Smith	1998	News	✘	✘	197
	WePS 1	2007	Web	✘	✘	3,489
	Day et al.	2008	News	✔	✘	3,660
	WePS 2	2008	Web	✘	✘	3,432
	WePS 3	2009	Web	✘	✘	31,950
Wikification	Mihalcea	2007	Wiki	✔	✔	7,286
	Kulkarni	2009	Web	✔	✔	17,200
	Milne	2010	Wiki	✔	✔	11,000
TAC	Cucerzan	2007	News	✔	✔	797
	Fader	2009	News	✘	✔	500
	TAC	2009	News	✘	✔	3,904
	TAC	2010	News, Blogs	✘	✔	3,750
	Dredze	2010	News	✘	✔	1,496
	Bentivogli	2010	News, Web, Tx.	✔	✔	16,851
	TAC	2011	News, Blogs	✘	✔	2,250
	Hoffart	2011	News	✔	✔	34,956
	TAC	2012	News, Blogs	✘	✔	2,226
	TAC	2013	News, Blogs, Fora	✘	✔	2,190

Table 3.1: Summary of NEL datasets.

annotating linked data and a description of a large, new whole-document annotated corpus in the news domain. The annotation campaign was part of the Computable News project, for full details, see Nothman (2014).

3.1 Cross-document coreference datasets

The seminal work on cross-document coreference resolution (CDCR) was published by Bagga and Baldwin (1998b). They performed experiments on a set of 197 documents from the New York Times whose text matched the expression `John.*?Smith`, where `.*?` is a non-greedy wildcard match up to the first instance of Smith, so only John Donnell Smith would be matched in John Donnell Smith bequeathed his herbarium to the

Smithsonian. The documents were manually grouped according to which John Smith entities they mentioned. None of the articles mentioned multiple John Smiths, so the only annotations were at the document level.

The John Smith dataset defines the problem as *one mention, many entities*: there are many entities that are referred to by an ambiguous name such as John Smith. However, there is another side to the problem: *one entity, many mentions*. An entity known as John Smith might also be known by other aliases (e.g. Jack Smith, Mr. Smith, etc.). In other words, there are both synonymy and ambiguity issues for named entities.

Most CDCR datasets are similarly collected by searching for a set of canonical name mentions, ignoring non-canonical references. For instance, Mann and Yarowsky (2003) collected a data set of web pages returned from 32 search engine queries for person names sampled from US census data. This data was later included in the WePS data described below. While ensuring that each document contains a canonical form for an ambiguous entity, this distribution of names may not match other tasks.

In contrast, Day et al. (2008) identify coreferent entity chains between documents in the ACE 2005 corpus (NIST, 2005), which already marks in-document coreference between proper noun, nominal and pronominal entity mentions. Marking in-document and cross-document coreference for all entities in a corpus addresses both synonymy and ambiguity issues.

Because manually annotating data is costly, there has been some interest in adopting the *pseudo-words* strategy of generating artificial word sense disambiguation (WSD) data, first described by Gale et al. (1992b). For WSD, the data is generated by taking two words that are not sense ambiguous, and replacing all instances of them with an ambiguous key. For instance, all instances of the words banana and door would be replaced by the ambiguous key banana_door. The original, unambiguous version is reserved as the gold standard for training and evaluation.

Cross-document coreference resolved data can be generated in the same way by taking all instances of two or more names, and conflating them under an anonymisation key such as Person X. The task is then to group the documents according to their

original name mentions. This strategy was first explored by Mann and Yarowsky (2003), and subsequently by Niu et al. (2004) and Gooi and Allan (2004).

Pseudo-data generation is problematic for both word sense and named entity disambiguation, but for different reasons. For WSD, most ambiguities are between related senses. For instance, the tennis and mathematical meanings of the word set can be linked back to a common concept. Few sense ambiguities are between unrelated concepts such as banana and door, and it is very difficult to select word pairs that reflect the meaningful relationships between word senses.

For named entity disambiguation, there is little reason to believe that two people named John Smith will share any more properties than one entity named Paul Simonell and another named Hugh Diamoni, so the criticism of pseudo-data that has been made about word sense disambiguation does not apply. On the other hand, named entities have interesting internal structures that a named entity disambiguation system might want to exploit. For instance, the use of a title such as Mr. or Dr. may be a critical clue and removing it may make disambiguation more difficult.

The first large data set for CDCR was distributed for the Web People Search shared task (Artiles et al., 2007). The data set consisted of up to 100 web search results for 49 personal names, for a total data set of 3489 documents manually sorted into 527 clusters. The task was repeated the following year, with a new evaluation set consisting of 3432 documents sorted into 559 clusters (Artiles et al., 2009). The most recent task, WePS-III, provides a list of 300 people and the top 200 web documents retrieved from the Yahoo! search engine. The task also includes extracting biographic details such as birth date and place, aliases and educational details. Evaluation of the clustering and attribution was manual and only two people per name were checked.

WePS-III also added an additional entity disambiguation task, targeted at Online Reputation Management. The organisers searched Twitter for posts about any of 100 companies, selected according to the ambiguity of their names—companies within names that were too ambiguous or too unambiguous were excluded, which may make for an unrealistic task, as it removes very easy and very hard cases for disambiguation.

```
<query id="EL11">
  <name>Abbot</name>
  <docid>LTW_ENG_20081022.0009.LDC2009T13</docid>
</query>
```

"Abbot [sic] and Costello: The Complete Universal Pictures Collection"

Figure 3.1: Example TAC query and excerpt from its document.

Mechanical Turk was used to cheaply determine which of 100 tweets per company name actually referred to the company of interest. Participants were supplied the tweets, the company name, and the URL of the company's homepage. This task is closer to named entity linking than cross-document coreference resolution, but shares a common weakness of CDCR data: the data was collected by searching for the company name, so the task does not address named entity synonymy.

3.2 TAC datasets

The TAC datasets are composed of queries (e.g. Figure 3.1) that specify a mention and a document in which it is found. A system must return the entity's ID in the KB or NIL. Since 2011, systems must also cluster NIL queries, supplying distinct NIL IDs. The organisers also distribute a TAC KB derived from 818,741 English Wikipedia articles from a October 2008 snapshot (Ji et al., 2010). Figure 3.2 shows a KB entry containing a name string, automatically assigned entity type, entity ID, list of slot name-value pairs from infoboxes and the text without markup.

The source corpora contain news and web documents from Gigaword 4[1], and discussion forum posts in 2013. During the creation of the dataset, the annotators did not select mentions randomly. Instead, they favoured "confusable" mentions which have many (more than seven) or no matches in the KB. These ambiguous queries provide a more challenging task.

[1] http://catalog.ldc.upenn.edu/LDC2009T13

```
<entity wiki_title="Bud_Abbott" type="PER" id="E0064214" name="Bud Abbott">
  <facts class="Infobox actor">
    <fact name="name">Bud Abbott</fact>
    <fact name="birthname">William Alexander Abbott</fact>
    <fact name="birthdate">October 2, 1895 (1895-10-02)</fact>
    <fact name="birthplace">
      <link entity_id="E0699919">Asbury Park</link>,
      <link entity_id="E0769300">New Jersey</link>
    </fact>
    <fact name="deathdate">April 24, 1974 (aged 78) <link></link></fact>
    <fact name="deathplace">
      <link>Woodland Hills</link>, <link entity_id="E0739132">California</link>
    </fact>
    <fact name="occupation">
      <link entity_id="E0239778">Actor</link>, <link>Comedian</link>
    </fact>
    <fact name="spouse">Betty Smith (1918-1974) (his death)</fact>
    <fact name="children">Bud Abbott, Jr, Vickie Abbott</fact>
  </facts>
  <wiki_text>
    <![CDATA[Bud Abbott

William Alexander ‘‘Bud’’ Abbott (October 2, 1895--April 24, 1974) was an
American actor, producer and comedian born in Asbury Park, New Jersey. He is
best remembered as the straight man of the comedy team of Abbott and Costello,
with Lou Costello.
  ...
    ]]>
  </wiki_text>
</entity>
```

Figure 3.2: Excerpt from the TAC KB entry for Bud Abbott.

D.	Qs	KB	NIL	KB %	NIL %	PER	ORG	GPE	NW	WB	DF
09	3,904	1,675	2,229	42.9	57.1	16.1	69.4	14.5	100.0	0.0	0.0
$10_{tr.}$	1,500	1,074	426	71.6	28.4	33.3	33.3	33.3	100.0	0.0	0.0
10	2,250	1,020	1,230	45.3	54.7	33.4	33.3	33.3	66.7	33.3	0.0
11	2,250	1,124	1,126	50.0	50.0	33.3	33.3	33.3	66.3	33.7	0.0
12	2,226	1,177	1,049	52.9	47.1	41.2	31.7	27.0	66.1	33.9	0.0
13	2,190	1,090	1,100	49.8	50.2	31.3	32.0	36.7	51.8	15.7	32.6

Table 3.2: Statistics for the TAC datasets: number of queries, split by query type, mention type and genre.

Table 3.2 shows basic statistics for the TAC datasets. This includes the total number of queries and broken down by type: KB or NIL. The TAC 09 dataset is substantially larger than the others, and also has the largest proportion of NIL queries. The TAC 10 $_{train}$ dataset is smaller, but is mostly made up of KB queries. All datasets since tend to be a similar size, around 2,250 queries and evenly balanced by type. The mention's entity type (PER, ORG or GPE) is skewed to organisations in TAC 09, and then balanced until TAC 12, where people are the largest class. Since 2010, the TAC evaluation datasets have started to include other sources such as websites (WB) and discussion forums (DF), but still mostly address newswire stories (NW).

Table 3.3 addresses ambiguity. We define mention ambiguity ($\mathbb{M}$), the mean number of entities per mention and entity ambiguity ($\mathbb{E}$), the mean number of mentions per entity. More concretely, a $\mathbb{M}$ and $\mathbb{E}$ of 1 indicate that the mapping between mention and entity is unambiguous. Most datasets have an even balance between $\mathbb{M}$ and $\mathbb{E}$, but TAC 12 has a much higher mention ambiguity than entity ambiguity, where TAC 13 reverses this pattern. The higher mean number of entities per mention is problematic for basic clustering approaches that cluster by mention strings.

We can also analyse how mentions related to the titles of KB entities. Table 3.4 shows the number of KB queries in each dataset and also the proportion of different types of match. We normalise both mention and title by case, and for the entity

Dataset	NIL clusters	$\mathbb{M}$	$\mathbb{E}$
09	✓	7.7	7.0
10_{train}	×	2.3	2.3
10	✓	3.0	2.6
11	✓	1.7	1.5
12	✓	2.9	1.1
13	✓	1.4	3.0

Table 3.3: Ambiguity in the TAC datasets: whether NILs are clustered, mean number of entities per mention ($\mathbb{M}$) and mean number of mentions per entity ($\mathbb{E}$).

title, removing content after a comma or in parentheses (e.g. we extract john howard from `John Howard, 1st Duke of Norfolk` and `John Howard (author)`). We count exact matches and cases where the mention is a substring of the normalised title. If the mention still does not match the title, we check if the mention is in upper case to identify acronyms or classify it as other. Exact matches are most common in the TAC 10 datasets. While this means that identifying the true entity as a candidate should be straightforward, this does not measure how ambiguous the mention is. Substring matches increase in TAC 11 and TAC 12, where more mentions only include part of the title. Acronyms were frequent in TAC 09 and TAC 11 and other cases have decreased in more recent datasets. The relatively high proportion of other in TAC 13 points to efforts to provide a more challenging task.

In summary, the TAC datasets have become more evenly balanced, and have tended to concentrate more on mention ambiguity and substring matches. While it is not clear what distribution of KB to NIL queries is realistic, and to what application, a well-balanced dataset makes it difficult for systems to gain advantage optimizing for one type or the other. Increasing mention ambiguity and substring matches mean that more sophisticated linguistic processing and disambiguation may be required for good performance.

Dataset	KB	Exact	Substring	Acronym	Other
09	1,675	43.9	14.5	18.3	23.3
10_{train}	1,074	70.3	11.4	6.8	11.5
10	1,020	60.8	13.5	7.0	18.7
11	1,124	46.9	24.9	19.1	9.1
12	1,177	46.0	41.2	7.0	5.9
13	1,090	38.4	15.5	8.6	37.4

Table 3.4: Match statistics for KB queries: total number and proportions of different matches between the mention and entity title.

3.3 Other datasets

In addition to the TAC datasets, other researchers have linked text to Wikipedia. Fader et al. (2009) use TEXTRUNNER to identify 500 predicate-argument relation tuples to use as mentions. These are drawn from a corpus of 500 million Web pages, covering various topics and genres, and the documents containing the tuples are included as document context. Considering only relations where one argument was a proper noun, the authors manually identified the Wikipedia page corresponding to the first argument, assigning NIL if there is no corresponding page. 160 of the 500 mentions resolved to NIL. Dredze et al. (2010) performed manual annotation using a similar methodology to TAC, in order to generate additional training data. They linked 1,496 mentions from news text to the TAC knowledge base, of which 270 resolved to NIL—a substantially lower percentage of NIL-linked queries than the 2009 and 2010 TAC data.

Other datasets are annotated to link all mentions to Wikipedia. Cucerzan (2007) manually linked all entities from 20 MSNBC news articles to a 2006 Wikipedia dump, for a total of 756 links, with 127 resolving to NIL. This data set is particularly interesting because mentions were linked exhaustively over articles, unlike the TAC data, where each query consists of one mention. The Cucerzan dataset thus gives a better indication of how a real-world system might perform on whole documents. However,

the Cucerzan data was collected by correcting the output of his system, which may marginally bias the data towards his approach. This may make the data unsuitable for comparison between systems.

There has been some work linking gold-standard mentions from NER corpora to Wikipedia. Hoffart et al. annotate 34,956 mentions[2] in 1,393 CoNLL-03 English articles, with roughly 80% mapping to a YAGO entity (and the equivalent Wikipedia and Freebase identifiers). Bentivogli et al. (2010) link mentions in coreference chains from ACE 2005 English data to Wikipedia. This is a subset of the Wikification task, as only concepts that appear in the same coreference chain as an entity are linked (e.g. President will be linked if it is clustered with Abraham Lincoln). This raises some issues, as they allow linking nominal mentions to multiple concepts of varying specificity (e.g. President may be linked to `President` and `President of the United States`). They report good agreement between their two annotators, 0.94 using a Dice metric. This approach is important since it builds on existing resources, and allows researchers to build on NER systems already tailored to the standard corpora. Moreover, gold-standard entity boundaries can factor out pipeline error from the NER process and allow evaluation of the linking component in isolation.

Fully-fledged wikification datasets are also of interest and include mentions linked to entities and general concepts in Wikipedia. Kulkarni et al. (2009) created the IITB[3] by annotating 107 news stories and linking 17,200 mentions to Wikipedia. They find a mention ambiguity of 5.3, but while this is a realistic estimate, it cannot be compared to ambiguity levels in TAC data. Firstly, TAC considers only entity names where the IITB considers general concepts as well. Also, the ambiguity rates in TAC are a consequence of the organisers' distribution of the data where the IITB data reflects real data. Milne and Witten (2008) take a different approach, using their system to automatically link 50 documents from the AQUAINT corpus and have human annotators check for

[2] Just shy of the 35,089 mentions listed in Table 2 of Tjong Kim Sang and De Meulder (2003).
[3] This dataset is named after the Indian Institute of Technology Bombay.

Q	The set of queries in the dataset
$\mathcal{G}$	Gold standard annotations for data set ($\|\mathcal{G}\| = \|Q\|$)
$\mathcal{G}_i$	Gold standard for query i (KB ID or NIL)
$G(q)$	The queries in gold-standard cluster for query q, including q
$S(q)$	The queries in system cluster for query q, including q
$\mathcal{C}$	Candidate sets from system output ($\|\mathcal{C}\| = \|Q\|$)
$\mathcal{C}_i$	Candidate set for query i
$\mathcal{C}_{i,j}$	Candidate at rank j for query i (where $\mathcal{C}_i \neq \emptyset$)

Table 3.5: Notation for searcher analysis measures.

correctness. While this process will find incorrect links, it will not propose correct links that the system missed.

3.4 Evaluation metrics

Our evaluation metrics closely follow those in TAC competition. We use the following evaluation measures, defined using the notation in Table 3.5. The first, accuracy (A), is the official TAC measure in 2009–2010 for evaluation of end-to-end systems.[4]

accuracy (A): percentage of correctly linked queries.

$$A = \frac{|\{\mathcal{C}_{i,0} | \mathcal{C}_{i,0} = \mathcal{G}\}|}{|Q|} \tag{3.1}$$

TAC also reports KB accuracy ($A_{\mathcal{C}}$) and NIL accuracy ($A_{\emptyset}$), which are equivalent to our candidate recall and NIL recall with a maximum candidate set size of one. The remaining measures are introduced here to analyse candidate sets generated by different search strategies.

Since 2011, TAC has evaluated NIL clustering performance using a coreference metric adapted for NEL. The B^{3+} score extends B^3 (Bagga and Baldwin, 1998a) to take cluster links into account (Ji et al., 2011).

[4]Macro-averaged accuracy (over entities) was reported in TAC 09, but micro-averaged accuracy, is the prevailing standard.

Gold	System
(a, b)	(a, d)
(c, d)	(c)
	(b)

Metric	Queries				Value
	a	b	c	d	
Precision	$\frac{1}{2}$	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{2}$	0.75
Recall	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0.5

Figure 3.3: Example of B^{3+} query clusters (2 gold, 3 system) and scoring (per-query and aggregated). For example, query c is clustered with only correct queries in system output ($\frac{1}{1}$), but the system misses c's co-clustered query (d) from the gold-standard ($\frac{1}{2}$).

B^{3+}: precision, recall and F. Assumes that $correct(q, q')$ returns 1 if two queries are correctly related, 0 otherwise.

$$P = \frac{\sum_{q \in Q} \left(\sum_{q' \in S(q)} correct(q, q') / |S(q)| \right)}{|Q|} \tag{3.2}$$

$$R = \frac{\sum_{q \in Q} \left(\sum_{q' \in G(q)} correct(q, q') / |G(q)| \right)}{|Q|} \tag{3.3}$$

$$F = \frac{2PR}{P + R} \tag{3.4}$$

The metric is based on judging relations between two mentions. Two entities are correctly related if they appear in the same cluster in the gold standard and system output and, crucially for NEL, link to the correct KB entry if the query is non-NIL[5].

Figure 3.3 shows an example of how B^{3+} is calculated for a set of query clusters. Intuitively, systems that tend to place queries into their own clusters preference precision, and systems that cluster all queries together are more recall-oriented. Since the TAC datasets tend to have multiple mention queries per entity cluster and vice-versa (although this changes year-on-year, see Table 3.3), B^{3+} is a reasonable evaluation metric. Moreover, B^{3+} aggregates performance on individual queries and is an intuitive extension of the accuracy-based measures that precede it.

We note at this point that the TAC evaluation depends entirely on structuring the dataset as queries. This removes the need for systems to locate the query term in

[5]We cannot check this condition for NIL queries and are satisfied if the queries cluster together.

the text and simplifies evaluation as it does not have to consider whether a mention has been correctly identified. Evaluations of whole-document linking often use gold-standard mentions rather than perform NER. This is justified where the focus of the work is disambiguation, but it cannot be considered a reliable evaluation of an end-to-end system that would have to recognise mentions as well as link them. Evaluation can be viewed different ways: macro-averaged KB and NIL concept link F score over documents (Kulkarni et al., 2009), or micro- and macro-averaged KB NE mention link precision and mean-average-precision (Hoffart et al., 2011), or bag of articles extracted for a document. This latter technique treats linking as a document-level tagging process and does not reward multiple correct links to the same KB entry nor punish multiple incorrect links. A comprehensive overview of these issues is reported in Pilz and Paaß (2012). We define several metrics that we use to analyse performance at different points *within* a system.

candidate count ($\langle C \rangle$): mean cardinality of the candidate sets. Fewer candidates mean reduced disambiguation workload.

$$\langle C \rangle = \frac{\sum_i |\mathcal{C}_i|}{|Q|} \tag{3.5}$$

candidate precision ($P_{\mathcal{C}}$): percentage of non-empty candidate sets containing the correct entity. Note that NIL queries have no correct entity.

$$P_{\mathcal{C}} = \frac{|\{\mathcal{C}_i | \mathcal{G}_i \in \mathcal{C}_i\}|}{|\{\mathcal{C}_i | \mathcal{C}_i \neq \emptyset\}|} \tag{3.6}$$

candidate recall ($R_{\mathcal{C}}$): percentage of non-NIL queries where the candidate set includes the correct candidate.

$$R_{\mathcal{C}} = \frac{|\{\mathcal{C}_i | \mathcal{G}_i \neq \text{NIL} \wedge \mathcal{G}_i \in \mathcal{C}_i\}|}{|\{\mathcal{G}_i | \mathcal{G}_i \neq \text{NIL}\}|} \tag{3.7}$$

NIL precision ($P_\emptyset$): percentage of empty candidate sets that are correct (i.e. correspond to NIL queries).

$$P_\emptyset = \frac{|\{\mathcal{C}_i | \mathcal{C}_i = \emptyset \wedge \mathcal{G}_i = \text{NIL}\}|}{|\{\mathcal{C}_i | \mathcal{C}_i = \emptyset\}|} \tag{3.8}$$

NIL recall ($R_\emptyset$): percentage of NIL queries for which the candidate set is empty. A high $R_\emptyset$ is valuable because it is difficult for disambiguators to determine whether

queries are NIL-linked when candidates are returned.

$$R_{\emptyset} = \frac{|\{\mathcal{C}_i | \mathcal{G}_i = \text{NIL} \wedge \mathcal{C}_i = \emptyset\}|}{|\{\mathcal{G}_i | \mathcal{G}_i = \text{NIL}\}|} \tag{3.9}$$

These metrics are the consequence of our framework for analysing NEL systems. They allow us to explore how component affect one another and form the basis for our analysis in Chapter 4.

3.5 Developing a whole-document NEL dataset

Finally, we report on issues creating an annotated corpus for NEL. A primary motivation for this is a joint project with Fairfax Media[6], a news organisation that publishes major metropolitan newspapers in Australia and New Zealand. The project goal was to explore how NLP can be used to create a structured data layer over their extensive unstructured text holdings. Part of this project involved annotating mentions in Sydney Morning Herald (SMH)[7] stories and linking them to Wikipedia.

The web-based tool and overall strategy for crowd-sourced annotation is described in detail in Nothman (2014), where it is applied to annotating news corpora for events. The annotation scheme, tool and datasets are products of the project team, rather than the author alone, and is used as an adjunct to the TAC datasets. We include the description of process, commentary of the issues involved in linked annotation and statistics of the dataset, allowing a comparison between the query-oriented TAC data and whole documents.

Our combination of student and crowd[8] annotators use a custom-built web interface. Annotators indicate mention boundaries and NE types, cluster mentions into coreference chains, and select the appropriate Wikipedia link for the coreference chain, or enter a NIL title. These are then available to other annotators, so that an entity that occurs in multiple stories, but not Wikipedia, can be linked to the same

[6] http://fairfaxmedia.com.au
[7] www.smh.com.au
[8] www.freelancer.com

Coarse	Fine
PER	Individual
LOC	Location
ORG	Organisation
MISC	Artefact, Event, Facility, Generic Language, Product, Work of art

Table 3.6: SMH corpus entity annotation scheme.

identifier (e.g. `John Smith (context)`). We deliberately do not give strict instructions on how to choose the NIL title, giving the annotators flexibility as in Wikipedia. We assign a fine-grained label from Table 3.6 to each mention, but where we train NER models, we use the coarse labels, similar to CoNLL-03.

We selected stories that were between 200 and 2000 words and filtered some sections (e.g. births, deaths and marriages and tenders). Average agreement between pairs of annotators in our pilot-phase cohort is reasonable. Cohen's kappa (Carletta, 1996) is 0.8–0.88 when annotating entity types and 0.85–0.89 linking entities to KB nodes and we consider the task sufficiently stable to offer to crowdsourced workers. Supervision of the crowdsourced workers was close during the second phase, as it was judged that, while we had established that high agreement was possible, spot checking and adjudicating difficult cases as they occurred was sufficient. Despite these manual safeguards, we suspect that a degree of noise is present in the annotation.

Table 3.7 shows the number of documents, tokens, mentions and proportion of KB and NIL mentions in the different splits of our dataset. We note that the NIL proportion is between that of the 20% of the CoNLL-03 and 40% of the IITB data. There are several potential explanations. Datasets that are newer than the KB may include entities that were not prominent at the time of the KB snapshot. Also, news from more diverse sources (Australian or general web, as in IITB) may not contain more entities that are locally notable, and not included in Wikipedia.

Split	Docs	Tokens	Mentions	KB	NIL	PER	ORG	LOC	MISC
TRAIN	1,592	1,007,141	58,147	74.1	25.9	39.7	24.8	21.9	13.6
DEV	201	129,947	7,716	73.9	26.1	37.1	24.2	23.7	15.0
TEST	201	126,068	7,113	74.5	25.5	38.8	28.6	21.5	11.1
TOTAL	1,994	1,263,156	72,97	74.1	25.9	39.3	25.1	22.0	13.5

Table 3.7: SMH dataset statistics.

Dataset	$\mathbb{M}$	$\mathbb{E}$
TRAIN	2.9	4.9
DEV	2.1	3.0
TEST	2.1	3.0
TOTAL	3.1	5.3

Table 3.8: SMH dataset ambiguity.

Table 3.8 shows the ambiguity statistics for each split and in total (these are calculated over the entire dataset, not averages of the figure for each split). Note that this is for KB mentions only as we did not make special effort to provide consistent NIL IDs and so is most similar to the TAC 10 training set. Ambiguity is substantially higher in the training set, which may reflect some sampling issue when splitting the dataset. Our dataset exhibits almost twice as much entity ambiguity as mention ambiguity. This may be an artefact of annotating all mentions in the document rather than picking one mention per document as in TAC. So, we may see a wider range of mentions for a particular entity: Howard in a headline, John Howard and Mr Howard in the body of the story. This ratio of ambiguity is different to the TAC datasets, which have always been at least 1.0, meaning exact balance or a skew to mention ambiguity.

We also examine the match statistics (see Table 3.9). Viewing the KB queries this way suggests that our whole document dataset is closest to the recent TAC datasets (TAC 11 and TAC 12). These have exact matches as the most common, followed by substring matches. There are relatively few acronyms compared with harder non-

Dataset	KB	Exact	Substring	Acronym	Other
TRAIN	43,064	46.0	32.9	5.6	15.4
DEV	5,702	46.9	31.5	5.1	16.5
TEST	5,297	43.6	34.2	6.1	16.1
TOTAL	54,063	45.9	32.9	5.6	15.6

Table 3.9: SMH dataset match statistics for KB mentions.

matches. One criticism of the TAC task is that its distributions (type, NE type, genre) are not representative of other tasks as TAC queries are chosen for their difficulty. Thus using metrics to comparing them with observed data is important in characterising how the queries differ and suggesting how a TAC system may perform in other tasks.

One reason for annotating news stories is to train a NER model that we use for our linking systems. We train the C&C NER tagger (Curran and Clark, 2003) on TRAIN and this performs on TEST at 80.15% precision, 78.55% recall and 79.34% F.

The annotation task was not straightforward. It requires annotation of the NE boundary and type, and coreference chains (we do not include common noun mentions), as well as searching our Wikipedia snapshot for KB entries. The extra complexity means that annotators require sufficient training, but crowdsourcing works reasonably well. Freelancer is an outsourcing marketplace and an alternative venue to Amazon Mechanical Turk (AMT) for hiring annotators. One advantage of Freelancer over AMT is that there is opportunity to retain good annotators for longer-term annotation and further training. The granularity of the KB can be an issue, with subtle differences between entities, for example `Great Britain` refers to the island, `United Kingdom` the modern state and `United Kingdom of Great Britain and Ireland` the state circa 1801. Equally, an annotator may not be able to tease apart a complicated structure of a company and its local subsidiaries. Deciding a canonical name for a NIL entity is also challenging and annotators can identify different disambiguation strings (e.g. `John Howard (filmmaker)` or `John Howard (director)`).

3.6 Summary

This chapter has reviewed datasets used for CDCR and NEL tasks, particularly TAC. We have tracked how the TAC datasets and metrics have changed over time to incorporate NIL clustering and its evaluation. We concluded with some observations on the process and challenges of annotating a linked dataset. In the next chapter, we use the metrics and TAC datasets to evaluate three seminal systems within our analysis framework. Later, we contrast the performance of a state-of-the-art linker on the TAC and our whole-document SMH dataset.

4 Benchmarking seminal NEL systems

> In 2002, Taylor appeared on the "Twelve Drummers Drumming" Christmas card in the "Twelve Days of Christmas" set sold at Woolworths to raise money for the NSPCC—alongside the "other" Roger Taylor, the drummer for Duran Duran.

Excerpt from the Wikipedia article for `Roger Meddows Taylor`, the drummer for Queen.

The previous chapters reviewed the wide range of NEL research, including different approaches evaluated using different datasets and methodology. We briefly introduced our framework for NEL in Chapter 2 and discussed existing approaches to linking with respect to it. This chapter revisits the framework and applies it to compare three seminal systems from the literature: Bunescu and Paşca (2006), Cucerzan (2007) and Varma et al. (2009). While NEL systems are commonly described in terms of separate search and disambiguation components,[1] very little analysis has been performed that looks at the individual effect of these components.

We implement the three systems within our framework and evaluate them on common data. This analysis focuses on the initial linking task defined in TAC rather than the NIL clustering extensions. As such, we present an evaluation on the TAC 09 and TAC 10 datasets. These datasets are widely used in TAC competition and in general linking research and are thus a good choice. Finally, this chapter contributes a detailed analysis of components, especially the interaction between components. Linking is

[1] McCallum et al. (2000) also describe a similar decomposition, motivated by efficiency, for the related task of clustering citation references.

complex and this chapter's contribution is a framework for better understanding the task that is grounded in seminal and innovative approaches from the literature. The analysis presented in this chapter is reported in Hachey et al. (2013) and I was responsible for the implementation of the systems and much of the analysis.

4.1 A framework for linker analysis

Our framework identifies three key components that systems use to link text to knowledge base entries. While we evaluate our systems on TAC queries, the framework can certainly be applied to tasks that link all mentions in the document, or those that include general concepts as mentions, such as wikification. The key insight is that systems tend to be implemented as a pipeline and an error in a component can prevent correct linking downstream. We examine components in isolation and within the pipeline, as components are strongly dependent on upstream components.

The core task of a NEL system is to link one or more mentions, given a document context, to a knowledge base (KB) entity node or NIL. This can be separated into three main components: extractors, searchers and disambiguators.

Extractor Extraction is the detection and preparation of mentions for linking. This varies for different tasks and while TAC specifies a mention string and document (later they also supply character offsets for exact identification), other variations of the task may require entity or concept recognition. Even when mentions are supplied, additional mention detection and preparation may be desirable because information about other entities in the text is useful for disambiguation. Extraction may also include other preprocessing such as tokenisation, sentence boundary detection, and in-document coreference. In-document coreference, in particular, is important as it can be used to find more specific search terms. For example, we may identify that ABC matches Australian Broadcasting Corporation or that Howard matches John Howard. Identifying the most specific form of a mention is a major advantage for the remainder of the linking task.

Searcher Search is the process of generating a set of candidate KB entities for a mention. Titles and other Wikipedia-derived aliases can be used at this stage to capture synonyms or common misspellings. Some aliases may be precise (e.g. the Wikipedia title `John Howard` or redirect `Johnny howard`), but often noisier aliases such as hyperlink anchors can provide higher recall as they reflect the way that authors generate mentions. An ideal searcher should balance precision and recall to capture the correct entity while maintaining a small set of candidates. This reduces the computation required for disambiguation, and will benefit methods that consider all candidates when linking a mention.

Disambiguator In disambiguation, the best entity is selected for a mention. We frame this as a ranking problem over the candidate set. This can use features from the mention's context; matches with the candidate information, such as Wikipedia article text; statistics from the KB; and compatibility with candidates of other mentions in the document. Features can be combined in different ways, from unsupervised feature combination to models trained with links from Wikipedia or TAC datasets. Recognising NIL links can be modelled explicitly using a NIL pseudo-candidate, separate classifiers, or thresholds, where a mention without a high-scoring match would be marked NIL. Ultimately, a good disambiguation component would identify the correct entity (including NIL) for a mention, but its performance is heavily dependent on the candidates generated in the extraction and search components.

Table 4.1 summarises the extraction, search and disambiguation of the three systems, listing the techniques used in each. While there are some techniques common to all, there are some important differences. The following subsections give more detail about how the approaches can be decomposed into the three components of the framework and report on our efforts to reimplement the systems from their descriptions in the literature. Some components are included in the systems that we submitted in competition in TAC. We provide brief descriptions of those components here and with more detail provided in Chapter 5.

Component	Technique	$\mathcal{B}$	$\mathcal{C}$	$\mathcal{V}$
Extraction	NER and in-document CR	N/A	✓	✓
	Acronym expansion			✓
Search	Titles, redirects, disambiguation titles	✓	✓	✓
	Hyperlink anchors		✓	
	Bold			✓
	Conditional, KB filter			✓
Disambiguation	Unsupervised reranking		✓	✓
	Supervised reranking	✓		
	Context/article similarity	✓		✓
	Context/category similarity	✓	✓	
	Article graph		✓	

Table 4.1: Comparative summary of seminal linkers. We list the different techniques in each components of the framework for Bunescu and Paşca (2006) ($\mathcal{B}$), Cucerzan (2007) ($\mathcal{C}$) and Varma et al. (2009) ($\mathcal{V}$).

4.1.1 Bunescu and Paşca (2006)

Bunescu and Paşca (2006) was one of the first systems to extend beyond the CDCR task to explicitly link person mentions against a KB. The authors use support vector machines (SVM) to rank for disambiguation. However, system performance has not been compared against subsequent approaches.

Extractor Bunescu and Paşca use capitalisation heuristics to identify which Wikipedia articles are about named entities. They frame the NEL task as to disambiguate the targets of article hyperlinks in other Wikipedia articles and, as such, do not use an extraction component. When we re-implement this, we use NER to identify mentions.

Searcher The search component for Bunescu and Paşca is an exact match lookup against article, redirect, and disambiguation title aliases (these are the hyperlink anchors of a link to the article from a disambiguation page). It returns all matching articles as candidates.

Disambiguator The Bunescu and Paşca disambiguator uses a Support Vector Machine (SVM) ranking model, using the SVM^{light} toolkit (Joachims, 2006). Two types of features are used. The first feature type is the real-valued cosine similarity between the mention context (a 55-token window centred on the mention) and the text of the candidate entity page (see Equation 4.1 below). The second feature type is a taxonomy kernel, using features from the Cartesian product of the candidate article's categories and the mention's context. Wikipedia categories can themselves belong to more general categories and the categories used include all ancestor categories to provide a more general semantic context. For example, `John Howard` belongs to `Category:Prime Ministers of Australia`, which belongs to `Category:Federal political office-holders in Australia`.

The taxonomy kernel can have substantial memory costs for a candidate with many specific categories. We also evaluate on the TAC data, which contains organisations and geopolitical entities, which increases the number of categories of interest. Thus, our implementations had to limit the number of categories used in the kernel. We used a frequency cut-off of 200 to filter rare categories. Selecting all ancestors is another source of kernel size, so we restrict this to the union of great and great-great grandparent categories as this performed best in preliminary experiments.

Bunescu and Paşca include a NIL pseudo-candidate in the candidate list, allowing the SVM algorithm to learn to return NIL as the top-ranked option when no good candidate exists. We do not include NIL pseudo-candidates since this decreased performance in our development experiments (-0.5% accuracy). As mentioned above, this also allows us to hold the NIL-detection strategy constant for all disambiguation approaches. The learner is trained on the development data provided for the TAC 10 shared task. It is important to note that the Bunescu and Paşca approach is the only one here that relies on supervised learning. The original paper derived training sets of 12,288 to 38,726 ambiguous person mentions from Wikipedia. Here, we use the TAC 10 training data, which has 1,500 total hand-annotated person, organisation, and geopolitical entity mentions. The small size of this training set limits the performance

of the machine learning approach in the experiments here. However, this also reflects the challenges of adapting supervised approaches to differently framed NEL tasks.

4.1.2 Cucerzan (2007)

Cucerzan (2007) uses evidence from the whole document to disambiguate entities. This is combined with in-document coreference to identify more specific mentions for search.

Extractor Cucerzan uses a hybrid NER tagger based on capitalisation rules, and statistics derived from web data and CoNLL-03 NER shared task data (Tjong Kim Sang and De Meulder, 2003). They use in-document coreference rules to match shorter (i.e. Howard) mentions to longer equivalents (i.e. John Howard) where they share the same entity type, and to match acronyms to an expanded form.

Our implementation uses the C&C NER tagger (Curran and Clark, 2003) trained on CoNLL-03 data to extract entity mentions from the text. Next, naïve in-document coreference is performed by taking each mention and trying to match it to a longer, canonical, mention in the document. These are expected to be longer, more specific and easier to disambiguate. Each mention is examined in turn, longest to shortest, to see if it forms the prefix or suffix of a previous mention and is no more than three tokens shorter. Upper-case mentions are considered to be acronyms and mapped to a canonical mention if the acronym letters match the order of the initial characters of the mention's tokens. Note that we do not require clustered mentions to share the same entity type, since we view identity as stronger evidence than predicted NE type.

Searcher For candidate generation, canonical mentions are first case-normalised to comply with Wikipedia conventions. These are searched using exact-match lookup over article titles, redirect titles, and disambiguation titles. Any alias with content after a comma or in parentheses is normalised to remove the extra content (e.g. `John Howard (Australian actor)` → `John Howard` and `Toronto, New South Wales` → `Toronto`).

In contrast to Cucerzan, we do not use link anchor texts as search aliases because we found that they caused a substantial drop in performance: -5.2% KB accuracy on the MSNBC corpus (Cucerzan, 2007) and approximately $10\times$ worse runtime.

Disambiguator Cucerzan build a *document context* from the candidates of all mentions in the document. Each mention's candidate list is then re-ranked by its compatibility with the global context, incorporating information from the whole document in each linking decision. Document context is an aggregation of the *contexts* and *categories* of each candidate article for all mentions in the document. Context is defined as the set of hyperlink anchors from an article where the link is in the first paragraph or the link is reciprocal (i.e. article a links to a' and vice versa). This can be considered as a specific vocabulary of related entities and general concepts for an article. Categories in this case are the union of the article's Wikipedia categories and titles of `List of...` articles that link to the candidate. To assign a candidate its score, the contexts are weighted by their occurrence in the mention context document and the score is a measure of its overlap (matching contexts and categories) with those from the rest of the document. We explain this in more detail in Subsection 5.1.5, but note that this method combines information from the mention's context, the candidate's categories and the Wikipedia article graph.

We evaluated our reimplementation against Cucerzan's MSNBC corpus, scoring 86.8% against his 91.4%. This variation may be due to several implementation differences: we use our own NER system; we did not use hyperlink anchors for aliases; we use a different list of categories due to Wikipedia change (our snapshot is from November 2009 not April 2006); we do not shrink source document context where no clear entity candidate can be identified for a mention. We observed that the evaluation was quite sensitive to small system variations, because the system tended to score either very well or rather poorly on each document. This is because information from each candidate is considered equal, regardless of whether it is likely to be correct. This means that a spurious, but well connected, candidate can distort the document context meaning that other incorrect related candidates receive higher scores.

4.1.3 Varma et al. (2009)

Varma et al. (2009) describe a system that uses a carefully constructed backoff approach to candidate generation and a simple text similarity approach to disambiguation. Despite the fact that it eschewed the complex disambiguation approaches of other submitted systems, this system achieved the best result (82.2% accuracy) at the TAC 2009 shared task.

Extractor The system first determines whether a query is an acronym (e.g. ABC). This is based on a simple heuristic test that checked whether a query consists entirely of upper-case alphabetical characters. If it does, the query document is searched for an expanded form by scanning the document for a sequence of words starting with the letters from the acronym, ignoring stop words (e.g. Australian Broadcasting Corporation or Agricultural Bank of China). No other preprocessing of the query or query document is performed.

Searcher Different candidate generation strategies are followed conditioned on whether the mention is an acronym or not. For *acronyms*, if an expanded form is found in the query document, then this is matched against KB titles. Otherwise, the original query string is used in an exact-match lookup against titles, redirect and disambiguation titles, and bold terms in the first paragraph of an article. For *non-acronyms*, the mention is first matched against KB titles. If no match is found, the mention is searched against the same aliases described above. The Varma et al. system for TAC 09 also used metaphone search (Deorowicz and Ciura, 2005) against KB titles for non-acronym queries. We omitted this feature from our implementation because Varma et al. reported that it degraded performance in experiments conducted after the TAC data was released.[2]

Disambiguator The Varma et al. approach ranks candidates based on the cosine similarity between the mention context and the text of the candidate article. Here, the mention context is the full paragraph surrounding the mention, where paragraphs

[2]Personal communication

are easily identified by heuristics in the TAC source documents. The cosine score ranks candidates using the default formulation in Lucene:

$$Cosine(q,d) = \frac{|\mathcal{T}_q \cap \mathcal{T}_d|}{\max_{m \in \mathcal{M}} |\mathcal{T}_q \cap \mathcal{T}_m|} \times \sum_{t \in \mathcal{T}_q} \sqrt{\text{tf}(t,d)} \times \left(1 + \log \frac{|\mathcal{D}|}{\text{df}(t)}\right) \times \frac{1}{\sqrt{|\mathcal{T}_d|}} \quad (4.1)$$

where q is the text from the query context, d is the document text, $\mathcal{T}_i$ is the set of terms in i, $\mathcal{M}$ is the set of documents that match query q, $\text{tf}(t,d)$ is the frequency of term t in document d, $\mathcal{D}$ is the full document set, and $\text{df}(t)$ is the count of documents in $\mathcal{D}$ that include term t.

This section has used our analysis framework to describe similarities and differences between the three systems. These systems capture a range of extraction techniques: NER, in-document coreference resolution and acronym expansion. The search component typically uses high-precision Wikipedia aliases to retrieve candidates. Disambiguation uses unsupervised and supervised contextual matching: with the article, with the KB categories and with the article's graph.

4.2 Evaluation

This section evaluates the diverse systems described above on common datasets allowing for direct comparison. Table 4.2 shows the results on TAC 09 dataset. In addition to the systems described above, we report three baselines, the median and maximum scores from competition. The NIL baseline returns NIL for every query and its score reflects the proportion of NILs in the dataset. The other baselines use exact matching of titles and redirects. The title+redirect baseline in particular is a strong baseline for this task, achieving a score 5.2% above the TAC median and 5.9% below the TAC maximum.

System	Accuracy		
	All	KB	NIL
NIL baseline	57.1	0.0	100.0
title baseline	71.0	37.2	96.5
title+redirect baseline	76.3	54.6	92.6
Bunescu and Paşca	77.0	67.8	83.8
Cucerzan	78.3	71.3	83.5
Varma et al.	80.1	72.3	86.0
TAC 09 Median	71.1	63.5	78.9
TAC 09 Maximum	82.2	76.5	86.4

Table 4.2: Results on TAC 09 for baselines, systems and literature.

System	Accuracy		
	All	KB	NIL
NIL baseline	54.7	0.0	100.0
title baseline	69.6	35.0	98.4
title+redirect baseline	79.4	60.6	95.0
Bunescu and Paşca (CosDAB)	80.1	67.1	90.9
Cucerzan (CosDAB)	81.0	71.1	89.3
Bunescu and Paşca	80.8	68.4	91.1
Cucerzan	84.5	78.4	89.5
Varma et al.	81.6	70.5	90.7
TAC 10 Median	68.4	-	-
TAC 10 Maximum	86.8	80.6	92.0

Table 4.3: Result on TAC 10 for baselines, systems and literature.

As in the TAC competition, our Varma et al. (2009) system performs best. The Cucerzan and the Bunescu and Paşca systems perform only slightly better than the title+redirect baseline, which does not use any disambiguation, and simply queries for exact matches for the mention string over the title and redirect fields. However, both systems would have placed just outside the top 5 at TAC 09. The competitiveness of the Varma et al. approach suggests that good search is critical to NEL and different disambiguators have less impact.

Table 4.3 shows the results on the TAC 10 evaluation dataset. We again include our three baselines, median and maximum scores, as well as adding variants of our systems that use simple cosine similarity disambiguators as in Varma et al. (2009). In contrast to TAC 09, the Cucerzan system is the most accurate at 84.5%, 2% lower than the maximum TAC 10 score (Lehmann et al., 2010). The TAC 10 data has an even balance of mention entity types (TAC 09 has 69% ORG queries), with fewer acronym mentions (15% to the 21% in TAC 09)[3]. This may account for some performance loss for the Varma et al. (2009) linker, where the specialised acronym processing will only benefit organisation mentions.

Table 4.4 contains accuracy scores broken down by genre (news or web) and entity type (ORG, GPE or PER). The first thing to note is that no approach is consistently best across genres and entity types. This suggests that system combination by voting or entity-specific models may be worth investigating. Next, the percentage of NIL queries varies hugely across genre and entity types. This is indicated by the performance of the NIL baseline, for example 73% of ORG newswire queries are NIL. In particular, the NIL percentage in web text is much lower than in news text for ORG and PER entities, but much higher for GPE entities.

There are two striking results about the behaviour of the title+redirect baseline. First, the system performs near perfectly on PER entities in news text (97.0%). In part, this is probably attributable to the editorial standards associated with news, which

[3]These include KB and NIL mentions where Chapter 3 reports only KB queries as we compare with the gold-standard title.

System	News			Web		
	ORG	GPE	PER	ORG	GPE	PER
NIL baseline	73	21	91	33	57	33
title baseline	73	51	91	50	75	72
title+redirect baseline	75	66	97	80	**77**	83
Bunescu and Paşca (CosDAB)	78	66	97	88	66	87
Cucerzan (CosDAB)	**81**	68	**98**	86	60	88
Bunescu and Paşca	77	64	97	88	72	**90**
Cucerzan	77	**83**	**98**	84	72	88
Varma et al.	78	68	97	**90**	69	87

Table 4.4: Accuracy of systems on TAC 10 genre and entity type subsets for baselines, systems and literature.

results in PER entities mentioned in news generally being referred to using canonical forms for at least one mention in a story. However, since the queries for the evaluation data set are not randomly sampled, it is not possible to quantify this observation. The second striking result is the fact that the title+redirect baseline outperforms all implemented systems on GPE entities in web text. This suggests that candidate generation is very noisy for these entities, which results in an especially difficult disambiguation problem. For ORG entities, systems with cosine disambiguators (including Varma et al.) perform the best in both news and web text. It is also interesting to note that there is very little variation in scores for PER entities, especially in news text.

Overall, our Cucerzan implementation is best for newswire, but does worse on web text. This holds for the cosine as well as for other disambiguators from the literature. This suggests that the Cucerzan search strategy is more suited for more formal text. This may be attributed in part to the searcher's use of naïve coreference and acronym handling, which are more accurate on text that follows the journalistic conventions for introducing new entities into discourse fairly unambiguously. For the

Cucerzan disambiguator, the poorer performance of named entity recognition on web text is also likely to have the effect of introducing more noise into the document-level vector representations.

4.3 Error analysis

Having compared the systems using the final evaluation metrics, we now concentrate on the impact of the different components and the relationship between them. The TAC data does not provide gold-standard entity mentions, although query strings should mostly match, and so we are unable to directly evaluate extraction performance. Instead, we assess the search and disambiguation performance using the accuracy and search metrics introduced in Chapter 3. Extraction performance is indirectly measured through its impact on the search and disambiguation.

We first examine search: the coverage of Wikipedia's alias sources, direct searcher performance, the importance of different extraction techniques, the impact of query limits and then some error cases. Then we address disambiguation: the importance of extraction, the effect of varying searchers and examination of error cases.

4.3.1 Analysing alias sources

Wikipedia articles contain a range of alias sources of varying quality. Table 4.5 shows the candidate count ($\langle C \rangle$), candidate precision ($P_{\mathcal{C}}^{\infty}$), candidate recall ($R_{\mathcal{C}}^{\infty}$), NIL precision ($P_{\emptyset}$) and NIL recall ($R_{\emptyset}$) for the different alias sources used on our development set, TAC 09. The first thing to note is the performance of the Title alias source. Title queries return 0 or 1 entities, depending on whether there was an article whose title directly matched the query. The candidate count of 0.2 indicates that 20% of all query mentions (KB and NIL) match at least one Wikipedia title. Matches for KB queries are high precision (83.5%) and low recall (37.2%). This means that systems may benefit from a simple heuristic that returns a candidate if its title matches the

Alias Source	$\langle C \rangle$	$P_{\mathcal{C}}^{\infty}$	$R_{\mathcal{C}}^{\infty}$	$P_{\emptyset}$	$R_{\emptyset}$
Title	0.2	**83.5**	37.2	68.1	96.5
Redirect	0.1	74.6	20.0	62.1	96.2
Link	4.2	55.7	**80.1**	**88.6**	59.5
Bold	1.6	45.1	48.8	71.7	67.2
Hatnote	0.0	42.6	1.2	57.7	**99.9**
Truncated	1.2	37.8	24.5	62.2	78.6
DABTitle	3.5	34.2	29.3	58.7	65.1
DABRedirect	2.7	34.0	18.9	57.9	77.3

Table 4.5: Search over individual alias fields (TAC 09).

Alias Source	$\langle C \rangle$	$P_{\mathcal{C}}^{\infty}$	$R_{\mathcal{C}}^{\infty}$	$P_{\emptyset}$	$R_{\emptyset}$	A
Title	0.2	83.5	37.2	68.1	96.5	71.0
+Redirect	0.3	79.4	54.6	75.0	92.6	76.3
+Link	4.2	56.2	81.7	90.2	59.4	
+Bold	4.7	55.7	84.8	90.6	55.1	
+Hatnote	4.7	55.7	84.8	90.6	55.1	
+Truncated	5.0	55.7	85.4	90.6	54.2	
+DABTitle	6.9	56.5	87.6	90.8	53.3	
+DABRedirect	7.2	56.3	87.8	90.7	52.5	

Table 4.6: Search over multiple alias fields (TAC 09).

mention. Titles matching performs well for NIL queries: only 3.5% of NIL queries incorrectly matched a title and 68.1% of cases returned no match, the best outcome.

Table 4.6 shows how the number of candidates proposed increases as extra alias sources are considered (cumulatively), and how much candidate recall improves. The addition of link anchor texts increases candidate recall to 81.7%, but also greatly increases the number of candidates suggested. The NIL recall drops from 92.6% to 59.4%, which means that at least one candidate has been proposed for over 40% of the NIL-linked queries. This makes some form of NIL detection necessary, either a threshold or a supervised model, as used by Zheng et al. (2010). Using all alias

Alias Source	$\langle C \rangle$	$P_{\mathbb{C}}^{\infty}$	$R_{\mathbb{C}}^{\infty}$	$P_{\emptyset}$	$R_{\emptyset}$
Title	0.2	83.5	37.2	68.1	96.5
+Redirect	0.3	79.4	54.6	75.0	92.6
+Link	2.4	56.2	76.5	87.6	63.8
+Bold	2.4	55.8	77.1	88.2	62.9
+Hatnote	2.4	55.8	77.1	88.2	62.9
+Truncated	2.4	55.8	77.1	88.2	62.9
+DABTitle	2.4	55.8	77.1	88.2	62.9
+DABRedirect	2.4	55.4	77.1	88.1	62.2

Table 4.7: Backoff search over alias fields (TAC 09).

sources produces a candidate recall of 87.8%, with a candidate count of 7.2 per query. The candidate recall constitutes an upper bound on linking KB accuracy. That is, there are 12.2% of KB-linked queries which even a perfect disambiguator would not be able to answer correctly. Many of these queries are acronyms or short forms that could be retrieved by expanding the query with an appropriate full form from the source document and we explore this below.

To illustrate the value of high precision matching, we construct a title-match system that returns an entity whose title matches the query, or NIL otherwise. This achieves 71.0% accuracy on TAC 09, which is a fairly strong baseline as half of the 35 runs submitted to TAC 09 scored below it. Expanding this system to also consult redirect titles improves it to 76.3% linking accuracy. Only five of the fourteen TAC 09 teams achieved higher accuracy. The other alias sources potentially return multiple candidates, so their utility depends on the strength of the disambiguation component.

One way to reduce the number of candidates proposed is to use a *backoff* strategy for candidate generation. Using this strategy, the most reliable alias sources are considered first, and the system only consults the other alias sources if a source returns no candidates. Table 4.7 shows the performance of the backoff strategy as each alias source is considered, ordered according to their candidate precision. The backoff model quickly reaches its maximum candidate count of 2.4 at a candidate

Searcher	$\langle C \rangle$	$P_{\mathcal{C}}^{\infty}$	$R_{\mathcal{C}}^{\infty}$	$P_{\emptyset}$	$R_{\emptyset}$
Bunescu and Paşca	3.6	56.3	77.0	86.6	62.7
Cucerzan	3.2	58.6	79.3	88.8	65.1
Varma et al.	3.0	59.8	81.2	90.9	66.4

Table 4.8: Performance of searchers from the literature (TAC 09).

recall of 77.1%, which trades off approximately 10% of recall against a candidate list almost a third the size when the fields are searched together (Table 4.6). The high-recall, small candidate lists would be well suited to a cosine similarity disambiguator, relying on a complex search that allows a simple disambiguation. Indeed, the extra interactions with a search index that the backoff requires may mean that a lightweight disambiguator is preferred.

4.3.2 Searcher performance

Having analysed the impact of different combinations of alias sources, we report on the search configurations used in our three systems (see Table 4.8). The first row describes the performance of our Bunescu and Paşca searcher, which uses exact match over article, redirect, and disambiguation title aliases. The second row describes our Cucerzan searcher, which includes coreference and acronym handling. As described in Section 4.1.2, mentions are replaced by full forms, as determined by coreference and acronym detection heuristics. The query terms are searched using exact match over article, redirect, and disambiguation titles, as well as apposition-stripped article and redirect titles. Finally, the third row describes our Varma et al. searcher, which replaces acronyms with full forms where possible and employs a backoff search strategy that favours high-precision matching against article titles that map to the KB over alias search. Alias search includes exact match over article, redirect, and disambiguation titles, as well as bold terms in the first paragraph of an article.

Searcher	$\langle C \rangle$	$P_{\mathcal{C}}^{\infty}$	$R_{\mathcal{C}}^{\infty}$	$P_{\emptyset}$	$R_{\emptyset}$
Cucerzan	3.2	58.6	79.3	88.8	65.1
− coreference handling	4.1	53.4	79.3	89.0	56.6
Varma et al.	3.0	59.8	81.2	90.9	66.4
− acronym handling	3.8	54.0	79.4	89.6	57.9

Table 4.9: Effect of coreference/acronym handling on searching (TAC 09).

The Cucerzan and Varma et al. searchers perform best. They both achieve candidate precision of close to 60% at candidate recall near 80%. This suggests that coreference and acronym handling are important. High precision is also beneficial: the Varma et al. searcher is slightly better in terms of candidate precision (+1.2%) and recall (+1.9%) possibly from the bold field and candidate count (−0.2).

4.3.3 The impact of extractors on search

Table 4.9 contains a subtractive analysis of coreference and acronym handling in searchers from the literature. Coreference resolution results in a lower candidate count (−0.9 for Cucerzan and −0.8 for Varma et al.), while acronym handling increases candidate precision (+5.2% and +5.8% for the two systems). For Varma et al., there is also an increase in candidate recall (+1.8%). This highlights the importance of using more specific mention forms where possible, as they are more likely to match the canonical names that occur in Wikipedia.

4.3.4 Searcher query limits

One way to improve disambiguation efficiency is to reduce the number of candidates that must be considered, while retaining the correct candidate in the list. Figure 4.1 plots the candidate recall of our searcher implementations against the query limit—the maximum number of results returned by the search index. All three linkers start with candidate recall under 60% and climb to their maximum at a query limit

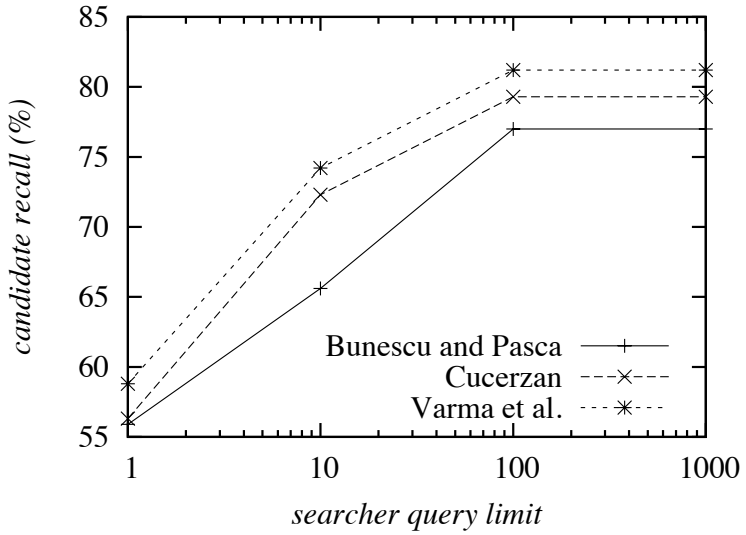


Figure 4.1: Effect of query limit on searcher candidate recall.

of 1,000. Higher limits give diminishing returns on recall, which suggests that a lower limit might increase efficiency (smaller candidate lists require less processing) at minimal cost to recall. However, going from a query limit of 100 down to 10 results in a substantial drop in candidate recall, especially for the Bunescu and Paşca searcher. Despite the possible efficiency gain, we use a limit of 1,000. This is essentially unlimited, which was the case in the original system descriptions.

4.3.5 Search errors

This section more closely examines the errors from searchers. Table 4.10 shows what contribution search errors make to overall performance. The search errors column counts the (necessarily KB) queries that are incorrect as the correct candidate could not be found. We also show the final number of incorrect KB queries for each system, which includes errors due to search and disambiguation. On average, 43% of KB accuracy errors are due to low-recall search. The final row shows the number of queries that all systems linked incorrectly, revealing a large proportion of difficult queries for search. Moreover, where systems make more errors in disambiguation

System	Search errors	Total errors
Bunescu and Paşca	386	899
Cucerzan	384	847
Varma et al.	316	776
Systems agree	287	301

Table 4.10: Number of KB accuracy errors due to search (TAC 09).

than search, few of these are shared indicating that their disambiguation components behave differently.

Table 4.11 shows the distribution of the common search errors, classified into broad categories. The Type column contains error totals over unique query mention strings, while the Token column contains error totals over individual queries. The most common type of search error occurs when a mention is underspecified or ambiguous (e.g. Health Department). Name variations—including acronyms (e.g. ABC), transliteration variations (e.g. Air Macao instead of Air Macau), and inserted or deleted tokens (e.g. Ali Akbar Khamenei intead of Ali Khamenei)—are also problematic.

The remaining cases are rare. There are a few cases that may indicate annotation errors. For example, several gold-standard articles are disambiguation pages. Other errors are due to targeting a mention at an incorrect point in an organisational structure. The distinction between general university sports teams and the teams for baseball, for example, is subtle and proved very difficult for the systems to draw. There are also some legitimate typographic errors: Blufton should be Bluffton.

We also investigated the impact of coreference on linking performance over a sample of 100 queries drawn at random from the TAC 09 data. Table 4.12 contains the counts of these queries that can be coreferred to a more specific mention in the text and the count that are acronyms. Among the 24 coreferrable queries, our Cucerzan coreference module correctly resolves five and our Varma et al. acronym expansion module correctly resolves six—three in common. Both systems correctly corefer some acronyms, including DCR $\mapsto$ Danish Council for Refugees and DMC $\mapsto$ DeLorean Motor

Error type	Examples	Type	Token
Ambiguous	Health Department, Garden City	20	118
Name variation	Air Macao, Cheli, ABC	26	109
Annotation	Mainland China, Michael Kennedy	6	38
Organisation	New Caledonia	5	14
Typographic	Blufton	4	8
Total	-	61	287

Table 4.11: Distribution of searcher errors on TAC 09 queries

Coreferrable	Acronym	Count
✔	✔	12
✔	✘	12
✘	✔	4
✘	✘	72

Table 4.12: Coreference analysis over 100 queries sampled from the TAC 09.

Co. The Varma et al. coreference additionally corefers more acronym cases such as CPN-UML ↦ Communist Party of Nepal (Unified Marxist-Leninist) and TSX ↦ Tokyo Stock Exchange. Since the Cucerzan implementation only corefers NEs, NE boundary detection error can rule out coreferring some acronyms, but correctly handles Cowboys ↦ Dallas Cowboys and Detroit ↦ Detroit Pistons. Note that while most acronyms are coreferrable (sometimes a longer form is not mentioned in the document), only half of the 24 coreferrable queries are acronyms, indicating that coreference is advantageous as it may identify a less ambiguous mention for the query.

4.3.6 Effect of extractors on disambiguation

Table 4.13 contains a subtractive analysis of coreference and acronym handling in disambiguators from the literature. In Table 4.9 above (effect of extractors on search), we saw that coreference and acronym handling reduces without significantly affecting

System	Accuracy		
	All	KB	NIL
Cucerzan	78.3	71.3	83.5
− coreference handling	74.9	69.4	79.0
Varma et al.	80.1	72.3	86.0
− acronym handling	77.3	69.7	83.0

Table 4.13: Effect of coreference/acronym handling on linking (TAC 09).

precision or recall. Here, we see that this results in substantial improvements in accuracy (A) of approximately 3%. For our Cucerzan implementation, the difference is mainly in terms of NIL accuracy, which sees a 4.5% increase due to the use of more specific name variants for search. Our Varma et al. implementation sees a more balanced increase in KB accuracy and NIL accuracy of approximately 3% each. The relatively large increase in KB accuracy for Varma et al. may be due to its search of the entire document for acronym expansions, rather than just other entity mentions as is the case for our Cucerzan coreference handling. This makes the acronym expansion less vulnerable to NER errors.

4.3.7 Effect of searchers on disambiguation

Table 4.14 contains results for versions of our Bunescu and Paşca and Cucerzan implementations that use the described candidate search strategies, but replace the disambiguation approach with the simple cosine disambiguator described in Section 4.1.3. The results here relate directly to the search results in Table 4.8 (comparison of implemented searchers), with high accuracy achieved by the searchers that have high candidate recall and low candidate count. In Table 4.8, the Varma et al. searcher outperforms the Bunescu and Cucerzan searchers in terms of candidate recall by 1.9% and 4.2% respectively, and in terms of candidate count by 0.2 and 0.6. Here, it also performs best in terms of accuracy at 80.1%–2.4% better than Bunescu and 1.3% better than Cucerzan.

Searcher	Accuracy		
	All	KB	NIL
Bunescu and Paşca	77.7	69.6	83.8
Cucerzan	78.8	69.7	85.6
Varma et al.	80.1	72.3	86.0

Table 4.14: Effect of searchers on cosine disambiguation (TAC 09).

Note that the Bunescu and Paşca and Cucerzan disambiguators (Table 4.2) perform worse than the cosine disambiguators defined in Subsection 4.1.3. This may be attributed in part to differences between the training and development testing data. For example, the distributions between NIL and KB queries changes as described in Table 3.2. Also, the TAC 2010 training data includes web documents while the TAC 09 evaluation data used for development testing here does not. For Bunescu and Paşca, the difference may also be due in part to the fact that the training data is fairly small. The held-out evaluation data, the TAC 10 evaluation data, is more similar to the training data. Results on this data (Table 4.3) suggest that the Bunescu and Paşca learning-to-rank disambiguator obtains higher accuracy than the corresponding cosine disambiguator (+0.7%), with a 1.5% increase in candidate recall.

4.3.8 Effect of swapping searchers

Table 4.15 contains a comparison of the Bunescu and Paşca and the Cucerzan disambiguators using the search strategy they describe and the search strategy from Varma et al.[4] For the Cucerzan system, we use Varma et al. search for the TAC query only and Cucerzan search for the other named entity mentions in the document. The results suggest that the high-precision Varma et al. search is generally beneficial, resulting in

[4]Note that the Varma et al. disambiguator corresponds to our cosine disambiguator. Therefore, the cosine disambiguation rows in Tables 4.2 and 4.3 correspond to the Bunescu and Paşca and Cucerzan systems with Varma et al. disambiguation. Note also that we do not swap in the Bunescu and Paşca searcher since it is not competitive (as discussed in Section 4.3.2).

Searcher	Disambiguator	Accuracy		
		All	KB	NIL
Bunescu and Paşca	Bunescu and Paşca	77.0	69.6	83.8
Varma et al.	Bunescu and Paşca	78.1	67.9	85.8
Cucerzan	Cucerzan	78.3	71.3	83.5
Varma et al.	Cucerzan	79.4	73.3	83.9

Table 4.15: Combinations of searchers on implemented disambiguators (TAC 09).

System	Disambiguator errors	Total errors
Bunescu and Paşca	513	899
Cucerzan	463	847
Varma et al.	460	776
Systems agree	14	301

Table 4.16: Number of KB accuracy errors due to disambiguation.

an increase in accuracy (+1.1%) for both the Bunescu and Paşca and the Cucerzan disambiguators. This suggests that selecting a good search strategy is crucial.

4.3.9 Disambiguator errors

Table 4.16 shows the number of disambiguator errors—queries in the TAC 09 data where the correct link was not returned because the disambiguator was unable to choose the correct candidate from the search results. It also shows the total number of KB accuracy errors (due to either searchers or disambiguators). The last row shows the number of queries for which all three systems return an incorrect result. The errors here account for the remaining errors (approximately 47%) that were not attributed to the searchers in Table 4.10 above. Interestingly, where search errors were largely common to all systems, few disambiguation errors are shared. While we do not explore it here, this suggests that voting may be able to combine system outputs if they are complementary.

Error type	Examples	Type	Token
Name variation	ABC, UT	2	14
Ambiguous	Garden City	4	10
Total	-	6	24

Table 4.17: Distribution of disambiguator errors on TAC 09 queries

	Type		Token	
System	Acronym	Not acronym	Acronym	Not acronym
Bunescu and Paşca	21	16	138	43
Cucerzan	30	33	81	115
Varma et al.	17	21	30	68

Table 4.18: Characteristic errors over TAC 09 queries

Table 4.17 shows a breakdown of common errors in the 100 query sample. The types of errors are less varied than search recall errors, and are dominated by cases where the entities have similar names and are from similar domains (e.g. sports teams called The Lions). Name variation still makes up a reasonable proportion of the errors at this stage, but these are exclusively acronyms (i.e. there are no nicknames, transliterations, or insertions/deletions as in the search errors above).

Finally, Table 4.18 summarises the counts of queries for which each system returned an incorrect entity while the other two did not. The errors are categorised according to whether the mention was an acronym or not, and counts are aggregated at type and token granularity. The relative proportion of acronym and non-acronym errors differs slightly for the three systems, with Bunescu and Paşca making more acronym errors, while Cucerzan balances the two, and Varma et al. makes more errors on non-acronyms. This reflects the level of acronym processing: Bunescu and Paşca has none whereas Varma et al. uses a finely tuned acronym search and Cucerzan (2007) uses coreference and some acronym handling. The token counts broadly follow the same trend, although skewed by the bursty distribution of types and tokens.

4.4 Summary

This chapter presents our framework for analysing NEL systems and compares three key systems from the literature. Decomposing systems into extraction, search and disambiguation reveals some surprising insights into the linking problem. Our detailed analyses show that successful search is critical for linking systems and baselines that take this into account are difficult to beat. The success of the title+redirect baseline underlines the importance of search for linking: using high-precision alias sources, using coreference resolution and acronym expansion to extract the most specific mention from the text, sets the upper bound of linking performance. In many ways, this is similar to the WSD problem, where a closed vocabulary and edited text obviate the need for search, and a most frequent sense baseline is very difficult to beat. The next chapter focuses on our submissions to the TAC shared task and describes a system that takes advantage of the insight and evaluation our framework provides.

5 TAC named entity linking

> ...this prison was planned by the penal reformer John Howard and Nash developed this into the finished building.
>
> `John Howard (prison reformer)` in `John Nash (architect)`

The Text Analysis Conference (TAC) hosts a number of evaluation workshops or tracks that promote and guide research into a specific NLP task or area. Shared data and metrics allow common, independent evaluation of different approaches. We participated in the English Entity Linking track in 2010, 2011 and 2012 (Radford et al., 2010, 2011, 2012) and our research has benefited immensely from the evaluation and task framework. Our NEL systems have evolved in response to the changing task guidelines, in pursuit of higher accuracy and to include other approaches from the literature. In Chapter 2, we discussed previous approaches to NEL and others in TAC.

This chapter reviews the task and introduces the principles that guide our system design—chiefly to link the whole document, not just the query. We then summarise the different components from our submitted systems and discuss our results from each year. We report on systems that accumulates our experience from participating in TAC and have state-of-the-art performance. The TAC metrics are oriented towards comparing systems, rather than introspective analysis. We discuss parameter tuning, feature impact and distribution of error types. We conclude with some discussion of the shared task environment and remaining challenges for NEL.

Our systems are developed in collaboration with other researchers in our lab and their contributions are as follows: Matt Honnibal developed the tuned search baseline

for Radford et al. (2010), Joel Nothman extracted data from Wikipedia snapshots, Glen Pink wrangled TAC data, Will Cannings extracted the Crosswikis data, Andrew Naoum generated aliases and Daniel Tse implemented the preliminary supervised system for Radford et al. (2012). The remaining work was conducted by the author.

5.0.1 Task description

We briefly review the NEL task within the scope of TAC and provide some more formal notation that we will use throughout this work. TAC linking is query based, a single term (called `name` in query input) in a document is matched to the knowledge base (KB). Our systems include some additional tests and we describe these below. Consider the query and the excerpt which it refers to in Example 1. Our NEL system should return the ID `E0064214` as the link for the term Abbot *[sic]*, matching the entry for the entertainer `Bud Abbott`.

(1) Also on DVD Oct. 28: "Abbot *[sic]* and Costello: The Complete Universal Pictures Collection";

```
<query id="EL11">
  <name>Abbot</name>
  <docid>LTW_ENG_20081022.0009.LDC2009T13</docid>
</query>
```

The query is difficult as the name is misspelled and there are multiple candidate entities whose name contains Abbot. Moreover, the reference could conceivably be to the man `Bud Abbott`, the comedy duo `Abbott and Costello` or the DVD boxset: `Abbott & Costello: The Complete Universal Pictures Collection`.[1] Although, given the mention string, the focus is clearly on just Abbott, but this does bring up the issue of nested mentions and how whole-document linking interacts with TAC queries.

Other queries refer to entities outside the KB and are known as NIL entities. In the case of Example 2, Abbas Moussawi is not in the TAC KB and so the appropriate answer is an ID beginning with NIL (e.g. NIL001). However, a matching entry is present in our

[1] www.amazon.com/Abbott-Costello-Complete-Universal-Collection

snapshot of Wikipedia from 2012 as `Abbas al-Musawi`. The different transliteration of his name illustrates the problem that entities have multiple aliases.

(2) In 1992, when Saguy was head of military intelligence, he recommended, at a Cabinet meeting, the elimination of Sheik Abbas Moussawi, leader of Hezbollah, a pro-Iranian, Lebanon-based Muslim fundamentalist organization.

```
<query id="EL1">
  <name>Abbas Moussawi</name>
  <docid>LTW_ENG_19960311.0047.LDC2007T07</docid>
</query>
```

The TAC evaluation from 2009 and 2010 considers how well systems link KB queries and identify NIL queries. More recently, the evaluation is more complex and requires systems to cluster NIL queries that refer to the same entity. Example 3 refers to the same `Abbas al-Musawi`, using a different transliteration.

(3) Hezbollah, Iran's main ally in Lebanon, said its guerrillas staged the onslaught to mark the assassination anniversaries of two top leaders of the group, Sheiks Abbas Musawi and Ragheb Harb.

```
<query id="EL3">
  <name>Abbas Musawi</name>
  <docid>APW_ENG_19950219.0048.LDC2007T07</docid>
</query>
```

We define our approach to the task more formally, with reference to terminology in Table 5.1 that we will use in this chapter.

TAC linking requires linking each query (t, d) in the dataset Q. The query supplies the term t and newswire or web document d in which it can be found. The document contains a number of named entity (NE) mentions m, which can be clustered together into coreference chains ($\mathbf{m}$) that refer to the same entity. Note that we only consider nominal mentions, rather than pronominal or common noun mentions as in full coreference resolution. The query term t can be matched against the chains and $\mathbf{m_q}$ identified—the query chain. This may involve finding a mention that matches the

Q	dataset of queries
(t, d)	TAC query, referencing a document (d) and term (t)
m	Named entity mention from the document d
$\mathbf{m}$	Chain of coreferred mentions in the document d
m_l	The longest mention in a chain
e	Candidate entity from the Wikipedia
$\mathbf{e_m}$	List of candidate entities for chain $\mathbf{m}$

Table 5.1: Linking terminology

term exactly or partially, then choosing the chain that contains the mention. If no mention matches the term, then we can create a mention from the term, adding it to its own chain. The KB contains many entity entries (e) and we can nominate a list of candidate entities ($\mathbf{e_m}$) for the query chain, $\mathbf{m_q}$, during search ($\mathbf{e_m}$ may be empty if there are no candidates).

The disambiguation process should identify the correct candidate for the query chain as the top candidate. We use a *Wikipedia mapping* approach to linking; we link to the full set of Wikipedia entities and then resolve linked entities back to their equivalents, if they exist, in the smaller TAC KB. Thus, for an entity that exists in both, we can map from the Wikipedia title to the TAC entity ID. We can return NIL if we fail to map to the TAC KB, or if a component identifies a NIL, or if there are no candidates for a mention. Where NIL queries are clustered, they should have the same NIL ID. With an understanding of the linking task as defined for TAC, we discuss our approaches in general and specific terms.

5.0.2 Design principles

We analyze NEL systems using the framework introduced in Chapter 2. First, entity mentions are extracted from the document. Then, for any mention, the KB is searched for plausible candidate entries. The reference is disambiguated, which can be seen as filtering and reranking the candidate list so that the top candidate is the most suitable link for the mention. Finally, NEL requires that NILs are classified, identifying

mentions that refer to entities outside the KB. With this scheme in mind, we outline some of the key principles that influence our system design and implementation.

Link the whole document The concept of *whole document linking* is central to our approach. The TAC entity linking task requires linking queries to the KB, where each query specifies a name (we resolve this to one or more mentions) and the document in which it is found. Our approach attempts to link all mentions in the document to the KB, then match the query mention to one of the mentions we have linked. This allows the linking of the query mention to take other mentions into account. We follow Cucerzan (2007) and Milne and Witten (2008) and link each mention independently, which should be distinguished from approaches that globally link all mentions in the document (Kulkarni et al., 2009).

Engineering matters We expect that linking every entity in the document will be advantageous for linking, but this can add significant overhead extracting, searching and disambiguating all entities rather than just the query entity. Consequently, engineering an efficient system is, for us, as much a necessity as a design goal. There are other benefits to efficiency: we can run experiments more quickly and incorporate more complex features. Considering the system in a real-world context is useful, as it prompts us to develop error analysis and KB curation tools, which are necessary in any commercial application, but benefit the research setting. Qualitatively, we found that high-level languages, such as Python, and efficient databases and low-level components, struck the right balance between performance and flexibility. A key component is our efficient document representation system, DOCREP (Dawborn and Curran, 2014). Any annotations are represented using offsets into the source document with more complex structures like entity tags and parse trees are layered over base token annotations. The format is programming-language independent and permits streaming, so that we can implement complex NLP pipelines easily.

Tracking pipeline error We implement our systems as a *pipeline* of components where we tokenize, identify NE mentions, cluster into coreference chains, search the KB and then disambiguate their candidates. Although this architecture is simple and

flexible, the system cannot recover from errors in an early component. For example, we may incorrectly extract a fragment of a name, perhaps only a surname, and retrieve a large list of spurious candidates. Equally, if we never retrieve the correct candidate entity, we cannot assign it as a link. For this reason, we track the recall and average rank of the correct candidate in the list of entities after every component has operated on the document. A good component should reduce the average correct rank, placing the correct candidate closer to the top of the list, but not reduce recall by discarding the correct candidate. This design allows fine-grained analysis of the pipeline's behaviour to help identify performance issues.

Wikipedia mapping Entity mentions are difficult to link in the absence of context. In these cases, the KB itself provides useful evidence for linking, such as the most likely entry for a given name and a larger and more detailed KB is advantageous. The TAC KB is a subset of a 2008 Wikipedia snapshot, containing only those entries with infobox data, as the dataset is used for the slot-filling track of the TAC shared task. We map these to a more recent Wikipedia snapshot that contains new articles and longer, richer revisions of existing articles. We discuss the implications of this later, but mapping from Wikipedia to a specialised, or reduced KB allows a system to take advantage of large-scale resources for linking.

Emphasise language Disambiguating entity mentions can require careful reading of context. Coreference can identify a longer, more specific mention, for example, the expanded form of an acronym, or a person's full name. If a sentence contains two interacting mentions, we might expect their relationship to be represented in the KB. We believe that concentrating on the language used to describe entities and their relationships in text is fundamental to linking them effectively.

We have reviewed the task and summarised the principles that guide our system design. In the next section, we describe the resources and methods we use for linking.

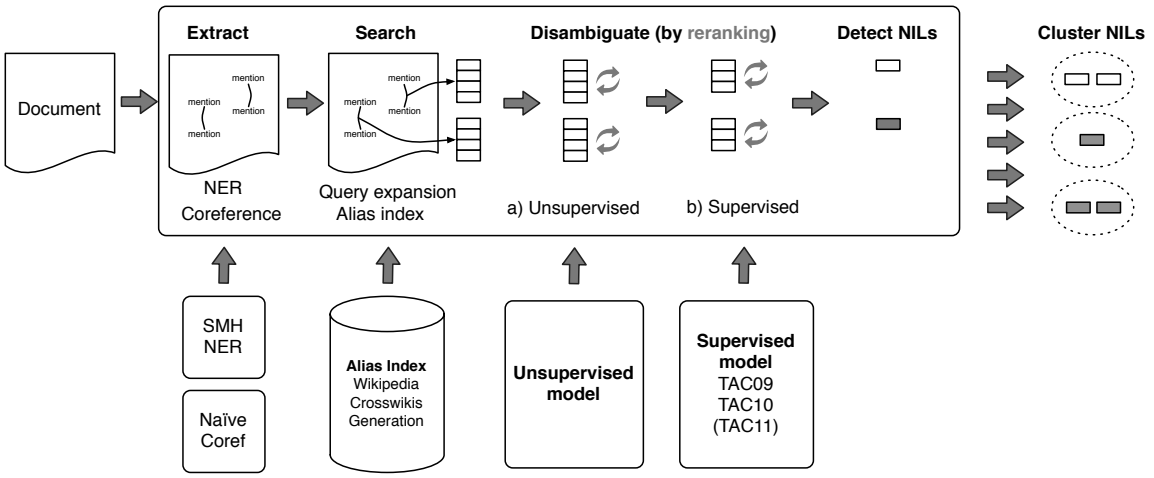


Figure 5.1: System diagram. This shows the progression of the document through the system (top) and main system components (bottom).

5.1 The internals of a named entity linker

Table 5.2 is a chronology of our TAC submissions. These include any resources used, methods used to extract mentions from the document, and techniques for searching candidate entries from the KB. We use several linking techniques to combine different features that model a link between an entity and KB entry, and methods for clustering NIL entities. Figure 5.1 shows architecture of our best system described in Section 5.3, but we include it as a guide to how the components are connected. The top half shows how the document passes through the extraction, search and disambiguation phases, including NIL thresholding and clustering. The bottom half sketches out the key components described below.

5.1.1 Resources

Named entity linking is data-intensive—large-scale data is a motivation for the task and key to the solution. Extracting and linking mentions from large corpora allows them to be collated by KB entry, providing an entity index for exploration. As a resource, large KBs like Wikipedia are a valuable source of information to inform

		Year		
		2010	2011	2012
Resources	Wikipedia 11/2009	✓	✓	
	Wikipedia 04/2012			✓
	Crosswikis			✓
	Alias generation			✓
Extraction	NER	✓	✓	✓
	In-document coreference	✓	✓	✓
Search	Alias index	✓	✓	✓
	Query expansion			✓
	Tuned search	✓		
	Alias reliability filtering	✓	✓	
Linking	Reranking pipeline	✓	✓	
	Unsupervised reranking			✓
	Supervised reranking			✓
Features	Cucerzan (2007) features	✓	✓	✓
	Wikipedia link graph reweighting	✓	✓	✓
	Alias matching			✓
	KB statistics			✓
	Context similarity			✓
	Entity type features			✓
	Title context			✓
NIL clustering	Rules	n/a	✓	✓
	Context clustering	n/a		✓

Table 5.2: Overview of our TAC systems.

Field	Example
Title	`John Howard`
Redirect	`John Howard (Australian politician)` `Johnny Howard`
Disambiguation page	John Howard (disambiguation)
Link anchor	John Howard
Bold	John Winston Howard
Infobox	`Political party = Liberal Party` `Spouse = Janette Parker`

Categories	`1939 Births`
	`Australian Anglicans`
	`Prime Minsters of Australia`

Table 5.3: Information from different fields of the Wikipedia article `John Howard`.

linking, providing much more than simply an entity's canonical name and description. Apart from the supplied data, the TAC KB and query documents, systems use a variety of other resources to help disambiguate entities. These can be stored as files where streaming access is required (i.e. source documents), key-value stores where we access entity data indexed by a canonical identifier (e.g. a title) or a full-text index for sophisticated search over different entity aliases.

Wikipedias The TAC knowledge base is derived from 818,741 articles, a subset of articles in a snapshot of English Wikipedia from October 2008 (Ji et al., 2010). Each entry is constructed from a Wikipedia article that contained an infobox and consists of a name string, automatically assigned entity type, entity ID, list of slot name-value pairs from infoboxes and the text without markup. Although derived from Wikipedia, the TAC KB pages lack several of its features that are useful for linking. We introduced these features in Chapter 2, but summarise them here in Table 5.3.

Redirect pages provide a useful source of entity aliases that include alternative names or common misspellings. Disambiguation pages explicitly list confusable

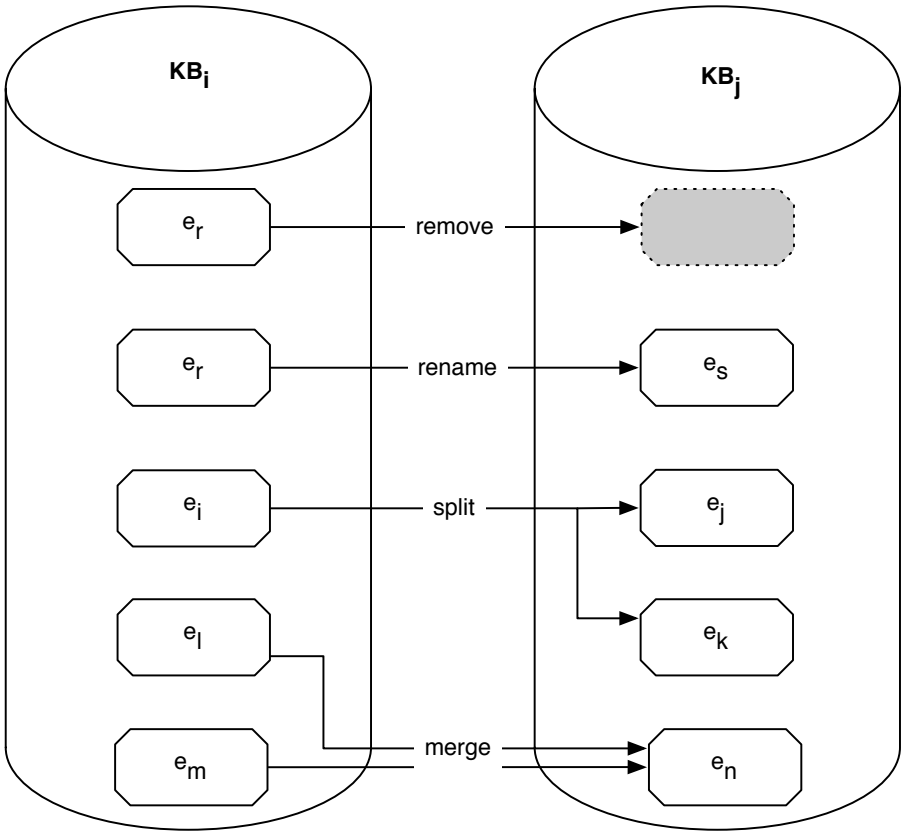


Figure 5.2: Operations mapping KB_i to KB_j: remove, rename, split and merge.

entities with text that describes their differences. Spans in bold font from the first paragraph indicate alternative canonical names. Links between articles induce an *article graph* where an article's neighbours can be interpreted as a set of related entities. Hyperlink anchor text can be aggregated to calculate a distribution over how entities are named as their articles are linked to. Structured data can be found in the infobox, which is heavily formatted by templates. Finally, categories are noisy indicators of an article's domain and provide a second degree association between any articles that share a category.

As mentioned above, we link to Wikipedia entries and map to the corresponding TAC KB entries. The TAC KB name string is the title of its source Wikipedia article and for each TAC KB entry, we can attempt to find the corresponding Wikipedia entry. Mapping the TAC KB to the Wikipedia snapshot it was derived from is trivial, but targeting later versions of Wikipedia presents several issues.

Mapping	Count	%
Exact	731,919	89.4
Renamed	69,270	8.5
Removed	17,552	2.1
TOTAL	818,741	100.0

Table 5.4: Statistics mapping from the TAC KB to our 2012 Wikipedia snapshot.

Figure 5.2 shows four possible changes. Articles may be removed, renamed split or merged in the newer version. This version may also include new articles, which map to NIL in the old KB and we do not include these here. Editors may decide that an article is not notable and should be removed. If we consider only one snapshot, it is impossible to link using the mapping technique as no Wikipedia entry exists for the TAC entry[2]. Assuming Wikipedia's editors retain a redirect to the new page (i.e. $e_r \rightarrow e_s$), renamed articles are easy to resolve. Where articles are split (i.e. $e_i \rightarrow e_j$ and e_k), the original title will presumably be retained (as a title or a redirect) to one of the new articles, effectively a renaming operation. The other articles, and indeed any new article, will have no equivalent in the older version and must be marked NIL. The final case is difficult as pages may be aggregated and if a system links to e_n in KB_j, should the link be e_l or e_m with respect to KB_k? For example, one query in the TAC 10 data should be linked to `Patrick "Tripp" Darling III` in the TAC KB, but in later versions of Wikipedia, that title redirects to `List of Dirty Sexy Money characters`. Having linked to the Wikipedia title `List of Dirty Sexy Money characters`, we must decide which TAC KB entry to finally return. Our strategy is to select the TAC KB entry whose title has the greatest token overlap with the longest mention, m_l.

We use two Wikipedia snapshots, from November 2009 (3,398,404 articles) and the other from April 2012 (3,704,351 articles). These are decompressed and parsed using mwlib[3] to extract different article components. MediaWiki markup allows arbitrary

[2]We could represent removed articles using "sentinels", but this would require tracking every removed article between the TAC snapshot and snapshot used for linking.

[3]http://code.pediapress.com/wiki/wiki/mwlib

HTML and templates that must be expanded before the content can be parsed. User-created markup inconsistencies and changes in editorial policy mean that extracting data from a snapshot is non-trivial and time-consuming (Nothman et al., 2013).[4] These are stored in a key-value store to allow quick access to an article's content indexed by title. We have experimented with a number of backends including Tokyo Tyrant,[5] Cassandra[6] and Hypertable.[7] Our current systems use Hypertable as we find this to be the most stable and efficient backend. We also use a Solr[8] full-text index for searching alias names as it is efficient for search and indexing and has a rich collection of analysers and query processors for experimentation. Table 5.4 shows the outcome for mapping each of the TAC KB entities to our Wikipedia snapshot. We can resolve approximately 98% of the entities to a Wikipedia article and find that 14,849 TAC KB entries are merged, just under 2%.[9] These statistics indicate that the Wikipedia mapping process has minimal effect on the linking evaluation—only a few entities in the TAC KB are unlinkable or require inverting the merging process.

Linking against Wikipedia and mapping to the TAC KB has implications for task realism. We discuss these in more detail in Section 5.4, but we make two basic assumptions. Firstly, using Wikipedia allows us to link using richer features such as unstructured and structured data from the article, and statistics drawn from the KB. Secondly, more recent versions of Wikipedia may be larger and more detailed than the target KB. These assumptions may not hold if the target KB cannot be aligned to Wikipedia, or if no larger, more detailed KB is available.

Crosswikis Earlier, we demonstrated the importance of pipeline error. One way this can manifest is in low recall search. If the correct candidate for a KB mention is not in the list, it cannot be linked to. Our 2012 system aims to increase recall to prevent these unrecoverable errors. Wikipedia is the main source of entity aliases: its redirect

[4] Joel Nothman extracted information from Wikipedia snapshots.
[5] http://fallabs.com/tokyotyrant
[6] http://cassandra.apache.org
[7] http://hypertable.com
[8] http://lucene.apache.org/solr
[9] This is not shown in the table, as merging affects exact and renamed entities.

pages, disambiguation pages and hyperlink anchors. All Wikipedia content is subject to its style guides, and while they are not universally enforced, aliases from outside Wikipedia may be more indicative of how entities are referred to in the news and web sources used for TAC. Other resources have extracted aliases from general web crawls: *Crosswikis* (Spitkovsky and Chang, 2012) and more recently *Wikilinks* (Singh et al., 2012). These external alias sources are less restrictive than Wikipedia, hopefully higher coverage, but noisier. Our 2012 systems use aliases extracted from Crosswikis, and stored in our full-text index to boost recall.[10]

Alias generation However large a web-scale resource is, it cannot directly generate aliases for novel entities. To solve this problem, we extract common transformations from an entity name to its corresponding redirect and generalise them into rules. These rules can generate missing aliases for known entities or generate variation aliases for an unknown entity. We use Levenshtein edit distance to identify common subsequences between string pairs, replacing rare words with wildcards. For example, given a title `Valve` and redirect `Valve Corporation`, we generate a rule that allows deleting a `Corporation` suffix to create an alias.

An automatically extracted set of 663,624 instances is manually filtered to select 434 high-frequency rules. Rules include transformations such as the removal of name titles, prefixes, suffixes and middle initials; the abbreviation and removal of organisation suffixes, state and country names. These are applied to the article titles creating new entity aliases.[11] A more sophisticated approach is to learn a model that can generate aliases (Andrews et al., 2012).

5.1.2 Extraction

Our systems mostly focus on whole document linking, detecting mentions and using in-document coreference to cluster them into *chains*. These tasks are key opportunities to use linguistically-aware methods and are central to our systems.

[10] Will Cannings extracted and stored these aliases.

[11] Andrew Naoum developed rules, applied them and stored the resulting aliases in the index

Named entity recognition The document is first tokenised and sentence boundaries detected. We tag entities mentions using the C&C Tools (Curran and Clark, 2003) with a 4-class (PER, ORG, LOC, MISC) model. Our 2010 system used a CoNLL-03 trained model, but from 2011, we use a model trained on the SMH Australian news corpus described in Chapter 3. The 4-class CoNLL scheme does not align exactly with the three-class scheme used for TAC, but there are no cases where we require a mapping between the TAC KB scheme and ours. We also do not include honorific titles in our entity labelling for PER entities. The query term is mapped to one of the mentions using the supplied byte offsets or by matching substrings of the mention tokens.

In-document coreference The TAC query term is not necessarily the most specific form of the entity's name in the document. We cluster all mentions into coreference chains using simple substring heuristics. This is a simpler variant of the coreference task as we do not consider pronominal or common noun coreference. Indeed, proper nouns are amongst the easier cases for coreference (Stoyanov et al., 2009) and so a simple approach is effective.

Mentions are processed longest first, and are added to an existing cluster or create one of their own. The algorithm normalizes mentions for case and removes honorific titles such as Mrs and organisation suffixes such as Corporation.[12] Exact matches to previous mentions are preferred (i.e. Ms Gillard or Gillard[13] matches Gillard), then non-upper-case unigram suffix matches (i.e. Gillard matches Julia Gillard), then non-upper-case unigram prefix matches (i.e. Julia matches Julia Gillard), then acronym matches where the initial upper-case characters (this restriction is relaxed for stopwords, which can be lower-case) of the NE (i.e. DoJ matches Department of Justice).

Assuming coreference chains are cheap to compute, linking chains rather than mentions has efficiency benefits, reducing the number of search queries and other processing dependent on the number of linkable items, and accumulating context across all mentions in the chain. There are some limitations of this approach. We

[12] These are manually collated from http://en.wikipedia.org/wiki/Business_entity.
[13] After honorific removal.

assume one distinct entity per document (Gale et al., 1992a) but this can lead to issues where our naïve rules merge mentions in error. People who share the same surname will be clustered together, as will companies and their subsidiaries, although we could develop heuristics to reduce this. Coreference is subtle, as Sydney in Geelong play Sydney in Sydney might ideally link to `Sydney Swans` and `Sydney` respectively. Despite these issues, our rules are robust and avoid the computational overhead of parsing that many state-of-the-art coreference resolvers require.

5.1.3 Search

Once a chain has been identified, the system must then retrieve a list of link candidate entities from the KB. Ideally, this list should be as short as possible and contain the correct candidate. Longer lists may be more likely to contain the correct entity, but at the cost of processing the other incorrect candidates. This can be especially expensive in approaches that seek to link the whole document or link jointly. Our methods aim for different balances of precision and recall.

Alias index We formulate a query to retrieve a chain's candidates from the alias full-text index. We index on fields from an entity's Wikipedia article: the *title* of the article, any titles that *redirect* to the article and hyperlink anchor text of their link in a Wikipedia *disambiguation* article. Our 2012 system focuses on high-recall search and uses improved preprocessing. All aliases are normalised for case, diacritics and unicode characters (Normalization Form Compatibility Composition)—prior to indexing and at query time. We rank based on the logarithm of the number of Wikipedia articles that link to the entity's article. We also search two noisier index fields: *crosswikis* and *generated*. We rely on our field values to provide variant aliases and use exact matching, although Solr allows fuzzier matches. So that matches on canonical names are ranked higher, we weight the *title* and *redirect* fields (weight=100) more than *disambiguation*, *crosswikis* and *generated* fields (weight=10). Matching the

query in the former fields will result in a greater score than a match in the latter. We limit the search results to 100 items as this seems reasonable, given Section 4.3.4.

Query expansion Another recall-oriented search technique is to add more document context to the chain's search query. Our first search query is the text of the longest mention (m_l) as we expect this to be well-specified. We maintain a list of backoff search queries to apply if there are no hits for the first search query. We exclude any single word NE mentions that are substrings of the longest mention. For example, if a chain consists of two mentions, Julia and Julia Gillard, we only search for the latter, as we make the assumption that it is more specific. If the TAC query term is not present in the search query terms, we add it to the backoff list. This guards against the case of pipeline error where other misrecognised mentions in the chain provide spurious search terms; at least matches for the original TAC query term will be retrieved.

Any state aliases to the right of a mention are expanded and added to the search query (i.e. Austin, TX will add Austin, Texas). Organizational suffixes such as Inc. and GmbH are removed and the resulting string added to the search query. The proliferation of bureaucracy poses difficulties for NEL since government departments have generic names that are ambiguous when the country is unknown. It is feasible, for example, for any country c to generate an entity `Department of Foreign Affairs, [`c`]`. Even if the official language of the country is not English, an English-language document may translate the department name for their readers, sometimes retaining the original language name. If country names are found in the document and any mentions start with Ministry, Department or Office, a search query of the mention and these country names are searched *first* and the original search queries added to the backoff list. This attempts to at least limit the set of candidates to bureaucracies of countries mentioned in the document.

We observed worse NER performance at the beginning of sentences where capitalized words were misidentified as NEs. To limit the impact of these pipeline errors, we add search terms that do not contain the first token in the sentence (i.e. we may add United Nations for Former United Nations).

Order	Field
1	Literal title (no apposition stripping)
2	Literal redirect title (no apposition stripping)
3	Bold words (all articles that contain a bolded term matching the mention)
4	Title (apposition stripped)
5	Redirect (apposition stripped)
6	Partial title match
7	Disambiguation (no apposition stripping)
8	Link anchor text (no apposition stripping)

Table 5.5: Tuned search fields.

Tuned search One of our 2009 systems follows from the DAMSEL system (Honnibal and Dale, 2009) and uses minimal disambiguation and so requires a search strategy tuned towards precision, rather than recall.[14] This is achieved by querying the alias fields according to their reliability, and stopping once a search query returns at least one candidate. If no candidates are returned, the next alias field is consulted. Table 5.5 shows the fields and their order.

It is important that these alias sources are consulted one by one. This prevents a candidate generated by a less reliable alias field from being ranked ahead of an article from a more reliable alias, such as title or redirect. Apposition stripping removes tokens after a comma or within parentheses to yield a minimal form of the title. A cosine similarity threshold of 0.01 is applied for all alias fields past the first, so an article must either have a title that matches the search query, or have text that is minimally similar to the source document. The order of the alias sources and the cosine threshold are determined experimentally on the TAC 09 data.

Alias reliability filtering A more general approach to precise search is to decide which of an entity's aliases are reliable. After the search step, each mention has a list of candidate entities. We filter this list to try and remove entities matched using noisy aliases. Aliases are normalised by case and are stripped of punctuation and a

[14] Matt Honnibal developed this system.

reliable alias is one that: matches a Wikipedia title, redirect or bold term; appears as a hyperlink anchor; is an acronym of another reliable alias of three or more words; or contains at least 50% of a words of another reliable alias. Any candidates that do not have the mention in their set of reliable aliases are discarded.

5.1.4 Disambiguation

Once a system has retrieved a list of candidates for a chain, it must disambiguate them. We treat this as reranking the chain's candidates, $\mathbf{e_m}$. Ideally, for a KB query, the correct candidate should be the highest scoring candidate in the list and we wish to discover positive evidence that indicates that this correct candidate should be linked to the chain. NIL queries pose a different problem as a system should use negative evidence for a link, the absence of any high scoring candidates, or features of the chain that may indicate its lack of notability. This subsection describes our disambiguation strategies at a high level and the next explains the features in detail.

Reranking pipeline Our initial approach to linking, submitted to TAC 10 and TAC 11, is based on a three-step pipeline where the candidate list is filtered for unreliable aliases, then ranked using the reimplementation of Cucerzan (2007). The final step reranks the candidates using the Wikipedia article graph.

Unsupervised reranking Our submissions for TAC 12 use a more flexible ranking strategy. The first is unsupervised and we manually select a set of features (based on their performance on the TAC 11 dataset) that are averaged for a final score. This method is similar to the heuristic approach taken by Lehmann et al. (2010), who use it as an initial step before more complex processing.

Supervised reranking The second strategy is to train a classifier on previous TAC queries.[15] There are two options to consider when training supervised linkers including what entity candidates should be used for training and should a NIL candidate be included. The resulting model can be sensitive to these choices and we found it surprisingly difficult to implement models during TAC that were more effective

[15]Daniel Tse implemented the preliminary features and classifier for our TAC submission.

than our simple unsupervised system. Despite this, the supervision is a more principled method for combining features and should better manage interactions between features than the unsupervised averaging.

The TAC 12 supervised system uses a log-linear regression model implemented with MegaM in binomial mode (Daumé III, 2004). This scores candidates on whether they should be linked to the query chain. We first apply the unsupervised linker above, then rerank the top 5 candidates using our supervised model, trained on TAC 09, TAC 10 training and TAC 10. This has two benefits—the reduced candidate set is less noisy, and the supervised features which are more costly to compute can operate over a focused set of candidates. During training, we take entity candidates for the query chain, extract features and learn weights from them. If the candidate is correct, it is assigned a LINK label and NOLINK otherwise. Our feature choice is substantially influenced by systems in TAC 11 (Anastácio et al., 2011; Zhao et al., 2011).

During linking, the candidate list is reranked by each candidate's score from the regression model. Note that NIL queries have no correct candidate in this framing of the learning task. Some systems include a NIL candidate (Bunescu and Paşca, 2006; Dredze et al., 2010) in their candidate list, but since there is no KB candidate, the features that can be generated must rely on document data and statistics about the candidate list itself. Their advantage, however is that learning NILs means that no thresholding need apply after prediction to decide whether a low-scoring top candidate indicates that the query should be linked to NIL. Our system includes any candidate from Wikipedia that we retrieve during search; we do not apply a NIL threshold and cluster NILs instead, as described in Section 5.1.6.

5.1.5 Features

We explained above how our systems rerank candidates, this subsection describes the features we use to model linking chains to candidates. Margin notes of the form

$\mathcal{C}$	Global categories—counts over all candidates
$\mathcal{C}[c]$	Global category count for category c
$\mathcal{C}_e$	Categories of the entity e
$\mathcal{T}$	Global contexts—set over all candidates
$\mathcal{T}_d$	Global document contexts—counts in the document
$\mathcal{T}_e$	Contexts for the entity e
$\mathcal{T}_d[t]$	Count of context t in the document

Table 5.6: Terminology used for Cucerzan (2007) features.

`feature_name` indicate a feature definition. We present a wide range of features and their motivation. Later results analyse their separate impact on linking performance.

Cucerzan (2007) features Whole document linking is a fundamental design principle and so we reimplement the Cucerzan disambiguation phase: scoring a candidate by the compatibility of its Wikipedia categories and contexts (hyperlink anchors) with those of other candidates in the document and context matches in the document. More specifically, we calculate a vector from a filtered set of categories and contexts aggregated for all candidates of the document's chains—global categories and contexts (see Table 5.6 for specific terminology). Note that here we interpret global as across the whole document and do not refer to joint optimisation.

A candidate's categories consist of Wikipedia categories except those whose name: contains one of the tokens (`article`, `page`, `date`, `year`, `birth`, `death`, `living`, `century`, `acronym`, `stub`); contains a four-digit number (i.e. a year); or is `Exclude in print`. Categories for a page also include the title of list and table pages that link to it. List and table pages are identified by looking for page titles that start with `List of` and `Table of`. A candidate's *category score* is the sum of global category counts for its categories, so candidates with categories that are common globally will be scored higher ($\sum_{c \in \mathcal{C}_e} \mathcal{C}[c]$). This is penalised subtractively by the number of entity categories ($|\mathcal{C}_e|$) to avoid rewarding pages with many categories. `cat_score`

inlinks(e, a)	List of hyperlinks to entity e with anchor text a
inlinks$(e, *)$	List of hyperlinks to entity e with any anchor text
inlinks$(*, a)$	List of hyperlinks with anchor text a
inlinks$(*, *)$	List of hyperlinks
$\{f : f \in$ inlinks$(e, *)\}$	Set of entities that link to e

Table 5.7: Wikipedia article graph terminology.

Contexts include text from parenthetical expressions in page titles (e.g. TV series from the title `Texas (TV series)`) and the anchor text of reciprocal links (links from `a` to `b` where `b` also links to `a`) and any links in the first paragraph of a page. We

`context_score` count how many times each global context actually appear in the document ($\mathcal{T}_d$). The *context score* is the sum of document occurrences for an entity's contexts terms ($\sum_{t \in \mathcal{T}_e} \mathcal{T}_d[t]$), so that entities with context matches in the document are rewarded.

The two scores are combined in the original Cucerzan formulation (Equation 5.1). The best candidate for a given mention m is the argmax of $s_{Cucerzan}$ over its candidates.

$$s_{Cucerzan}(e) = (\sum_{c \in \mathcal{C}_e} \mathcal{C}[c]) - |\mathcal{C}_e| + \sum_{t \in \mathcal{T}_e} \mathcal{T}_d[t] \tag{5.1}$$

One criticism of this feature is that it is robust to noisy search as all candidates are weighted equally, regardless of how spurious a match they might be. Thus a well-connected spurious candidate (e.g. `United States`) or a long document with many mentions can distort the document context, as its categories and contexts can overwhelm those of correct but less notable candidates. In the unsupervised and supervised models, we separate the calculated score into two components: category score and context score. These are normalised by the global total of categories and contexts respectively.

Wikipedia article graph reweighting The features above use the Wikipedia article graph via its contexts, but we also explicitly use the graph to disambiguate candidates. Table 5.7 shows some terminology used to calculate graph features.

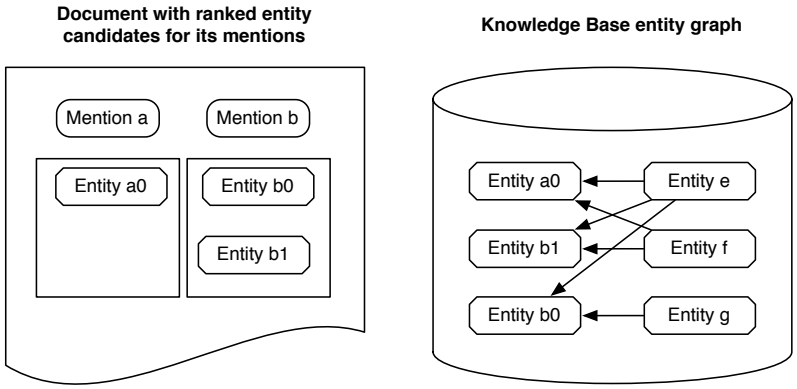


Figure 5.3: Link graph reweighting sketch, showing a document with mentions and entity candidates, and a Knowledge Base showing links between entities.

We hypothesise that a good candidate for a chain is well connected to the article graph of other chains in the document. We assume the candidate lists are ordered by some baseline metric, we use the ranking from the searcher, so that the top candidate for each chain has some chance of being correct. We can examine the hyperlinks that link to a candidate's article and identify a set of related entities. If we take the union of these entities for all top-ranked candidates of all chains in the document, we can approximate the document's neighbourhood in the Wikipedia article graph. We calculate the size of the intersection between a candidate's inlink set and those of the top candidates of other chains in the document; and use it to weight the candidate. Figure 5.3 motivates this technique, showing a document, two mentions and their respective lists of candidates with some given ordering (indicated by the label's integer suffix). The KB appears on the right hand side and shows a section of the article graph including the candidate entities (a0, b0, b1) and others that link to them (e, f, g). The initial ranking has b0 ranked above b1, however, the b1 shares more inlinks with a0 than b0 and should be boosted.

Assuming some initial reranked candidate list, we calculate the reweighted score using Equation 5.4. An entity's *top overlap* is the logarithm of the size of the intersection of its inlink set with the union of the inlink sets of all top-ranked entities of other chains ($\{\mathbf{m} \in d \setminus \mathbf{m}_e\}$). We add one to the intersection size to avoid logarithm

`top_overlap`

domain errors and one to the resulting logarithm to ensure that the score is always greater than one. The *top overlap* is then multiplied by the entity's score to boost it.

$$others(d, \mathbf{m_e}) = \bigcup_{n \in \{\mathbf{m} \in d \setminus \mathbf{m}_e\}} \{g : g \in \text{inlinks}(n, *)\} \quad (5.2)$$

$$\text{top\_overlap}(e) = \log\left(|\{f : f \in \text{inlinks}(e, *)\} \cap others(d, \mathbf{m}_e)| + 1\right) + 1 \quad (5.3)$$

$$(5.4)$$

Some wikification systems take advantage of unambiguous or reasonably ranked candidates to label general concepts and entities. Milne and Witten (2008) rank candidates by their average relatedness to unambiguous links, where relatedness incorporates inlink intersection. As we only consider entities and not general concepts, we may be less likely to find unambiguous chains and do not depend on doing so.

This is similar to the TAGME system (Ferragina and Scaiella, 2010), where candidates accrue votes from other chains and unambiguous chains are not assumed. Their relatedness is weighted by commonness (we call this reference probability, as explained below) and is calculated between the candidate and all other candidates in each voting chain. This approach is relatively computationally expensive, although they use a sliding window over mentions to limit the number of voting chains. Our approach prioritises efficiency at the expense of a precise calculation as we only consider the top candidates as context. This lightweight assessment of context allows us to efficiently incorporate the KB graph context for the whole document.

Alias matching If a chain refers to a candidate entity's name, we consider this a strong signal for linking. However both the chain and candidate can have multiple aliases; the chain is composed of one or more mentions and the candidate can have many aliases (i.e. redirects). The encyclopaedic style used in Wikipedia can lead to mismatches in article title and aliases commonly used to refer to them, for example, news stories often use UN to refer to the `United Nations`. Hence, we propose measures that take all aliases into account.

The *alias cosine* feature is the maximum similarity between all pairs of the chain's mentions (m $\in$ **m**) and its candidates' aliases ($a \in$ e), shown in Equation 5.5.

$$\text{alias\_cosine}(\mathbf{m}, e) = \underset{m \in \mathbf{m}, a \in e}{\text{argmax}}\, \text{sim}(v_m, v_a) \tag{5.5}$$

Similarity in this case is the cosine similarity of character bigram vectors created from the mention (v_m) and alias (v_a). Character bigrams are generated from a string by counting character pairs using a sliding window, for example abab would generate the bigrams ab, ba, ab. Cosine similarity is calculated over bigram vectors in the usual manner: $\text{sim}(u, v) = \frac{u \cdot v}{||u||\,||v||}$. `alias_cosine`

The value of this feature will be 1 if there are any mentions that are candidate aliases and should still provide some similarity in the case of close matches. We also use the score returned from the full-text alias index, Solr. The KB *score* feature is a combined indication of how well the chain matches the weighted fields (we weight title and redirect matches more than disambiguation, crosswikis and generated fields). `kb_score`

Our supervised system includes some features from other supervised TAC systems (Anastácio et al., 2011; Zhao et al., 2011). We calculate the similarity between m_l and the article title for the metrics: *Dice*, *cosine similarity*, *Levenshtein edit distance* and *Jaro-Winkler distance*. We also compute a group of binary features that are true if the article title *starts with* or *ends with* m_l, or if the title *contains* m_l and vice versa. `ttl_cosine` `ttl_dice` `ttl_jw` `ttl_starts` `ttl_ends` `ttl_contains`

KB statistics We can also use the structure of the KB itself to directly inform linking. Given two plausible candidates for a chain and limited context, we may prefer to use each candidate's prominence to inform the linking decision. We use the Wikipedia article graph to calculate the *kb prior* feature (5.6). `kb_prior`

$$\text{kb\_prior}(e) = p(e) = \frac{|\text{inlinks}(e, *)|}{|\text{inlinks}(*, *)|} \tag{5.6}$$

This is the number of links referring to the mention, normalised by the total number of links in the graph.

We can characterise how usual it is to refer to a particular entity using a particular
kb_refprob name in the KB. This is modelled using the *reference probability* feature—the conditional probability of a link to an entity given an anchor text in the article graph (5.7). This is a fundamental feature in wikification, often known as "commonness" of a sense (Milne and Witten, 2008). To calculate this for the chain of mentions, we use the longest mention assuming that it is the most informative. The value is thus the count of hyperlinks to e with the anchor $\mathrm{m_l}$ normalised by the count of all hyperlinks with the anchor $\mathrm{m_l}$.

$$\text{kb\_refprob}(\mathbf{m}, e) \equiv p(e|\mathrm{m_l}) = \frac{|\text{inlinks}(e, \mathrm{m_l})|}{|\text{inlinks}(*, \mathrm{m_l})|} \tag{5.7}$$

The *chain reference probability* calculates the metric using *all* mentions in the chain
chain_kb_refp. as per Equation 5.8.

$$\text{chain\_reference\_probability}(\mathbf{m}, e) = \sum_{m \in \mathbf{m}} \frac{|\text{inlinks}(e, m)|}{|\text{inlinks}(*, m)|} \tag{5.8}$$

Context similarity The similarity between the document and candidate article is an
unigram_cos. important factor for linking. We model this context at a document and sentence level, creating increasingly local similarity features. To process the query document, we extract bags of n-grams from the whole document or from each sentence that contains
bigram_cos. one of the chain's mentions. The whole Wikipedia article is transformed into a bag of n-grams. We normalise for case and encoding and remove stopwords and use cosine similarity between n-gram counts. We define features for the *unigram cosine similarity* and *bigram cosine similarity* at the document level and *sentence context*—
sent_cont unigram cosine similarity at the sentence level.

Token-based similarity can suffer from sparsity as a query document and a Wikipedia article can have similar context, but share few words. Following Anastácio et al. (2011), we use Latent Dirichlet Allocation (Blei et al., 2003) to model topics in both datasets so that we can capture similarity in a lower-dimensional space. We train

a model using the Vowpal Wabbit online machine learning toolkit,[16] with training parameters $k = 100$ (the number of topics), $\alpha = 1$ and $\rho = 0.1$. We use a corpus made up of documents from TAC 09 queries and the Wikipedia articles from our April 2012 snapshot. The *topic similarity* feature is the Hellinger distance (Kailath, 1967) between the predicted topic distribution of the query document and entity article, both using stemmed tokens. `topic_sim`

Modelling the similarity between document and article is a common technique (Bunescu and Paşca, 2006; Anastácio et al., 2011; Zhao et al., 2011) in linking and wikification systems, but approximate similarity can result in high scores for the related but not correct candidates, for example a university and its sporting teams.

Entity type features A chain and its correct linked entity should have the same entity type. Each mention in the chain is labelled in a 4-type scheme during the extraction phase. The Wikipedia articles are also assigned an entity type using a supervised classifier (Nothman et al., 2013). This uses the NE scheme used for the SMH corpus (see Table 3.6). Type matching features are commonly used for NEL systems (Anastácio et al., 2011; Zhao et al., 2011; Lehmann et al., 2010) and we adjust these to account for pipeline error. NE *type match* is a binary feature that is set if the article type matches the longest mention type after mapping from the larger scheme. `type_match`

We generate features from the combination of the article type and each mention type in the chain. For example, one chain may contain the mentions Paris Hilton and Paris labelled PER and incorrectly LOC. When matching against the candidate `Paris Hilton`, classified INDIVIDUAL, we generate pairwise features (PER, INDIVIDUAL) and (LOC, INDIVIDUAL), as well as a longest-mention feature for (PER, INDIVIDUAL). This models labeling noise and also lets a classifier learn weights for the correspondence of features from different schemes. `type_combo`

In addition to general type matching, we have a feature specific to person entity linking: PER *name match*. This is a binary feature that is true if any of the mentions in the chain are labelled PER, have two or more tokens and match the tokens of article `per_match`

[16] http://hunch.net/~vw

title before a left parenthesis or comma. For example, the feature value is true with a mention John Howard labelled PER and an entity candidate `John Howard (Australian actor)` or `John Howard`. This feature models a precise match between a person name mention and a candidate person name.

Location entities can be mentioned as the place in which some event occurs. The
`loc_prep` *follows* LOC *preposition* feature is binary and has the value true if a mention in the chain is labelled LOC and is preceded by a locative preposition:

> above, across, along, around, at, below, beneath, beyond, from, in, inside, into, near, on, onto, outside, over, through, throughout, to, toward, under, underneath, within

`ttl_context` **Title context** This is a measure of compatibility with titles of other entity candidates in the document. In the sentence The team toured Ontario, starting in Melbourne., Melbourne refers to `Melbourne, Ontario` rather than the more prominent Australian city `Melbourne`. If the entity `Ontario` is a candidate for another chain in the document, it should reinforce `Melbourne, Ontario` as a candidate. First, we extract the context from each candidate of each chain to try to identify *context* (e.g. `Ontario` in Melbourne, Ontario). Context here refers to non-parenthesized tokens after a comma in the candidate title. Then, we check to see if that context matches the title of any other candidate to identify *supporting entities* (e.g. `Ontario`).

Each entity's supporting entities can be sorted by the distance (number of sentences) from the entity. Each supporting entity is scored 1 if there is a supporting entity with an extra bonus point for being the closest and a further point for being in the same sentence. As such, `Melbourne, Ontario` would be scored 3 since it is the closest match in the same sentence supported by the candidate `Ontario` for the chain containing Ontario. This scoring scheme was developed using the TAC 11 data and rewards evidence close to the mention.

We have explained the features used in our TAC submissions. Table 5.8 lists the features used in our unsupervised and supervised TAC submissions.

Linking	Features
Unsupervised	cat_score, context_score, top_overlap, kb_prior kb_refprob, sent_cont, ttl_context
Supervised	kb_prior, kb_refprob, alias_cosine, ttl_sim ttl_sub, acro_match, unigram_cosine, topic_sim type_match, loc_prep

Table 5.8: Features used in our TAC 12 linkers.

5.1.6 NIL clustering

The 2009 and 2010 TAC tasks are limited to linking KB queries and identifying NIL queries. The task in 2011 and subsequent years includes clustering NIL entities, making the task more challenging, but also a more realistic test for the KB population task where new entities must be clustered and added to the KB.

Our approaches place more emphasis on linking than clustering, as we reason that the former is the easier task. This is because KB entries provide more evidence about an entity than isolated document mentions. For example, given two documents describing different phases of a person's life, they may not be similar to one another but would both be similar to passages of a biographic KB entry. Our naïve methods apply after linking, only to those chains without candidates. We use different heuristics to cluster their queries into distinct clusters.

Rules Our initial techniques use naïve rules to cluster queries that had been NIL linked (we do not re-cluster queries linked to a TAC KB entry). The rules are a sequence of one or more increasingly ambiguous attributes that queries are compared by. Clustering a query by its linked Wikipedia title is well-specified, whereas clustering by the query term is more ambiguous—essentially the situation prior to linking.

Attributes: W, t All queries that have no candidates or have their top candidate outside the TAC KB are considered NIL. Any NIL queries sharing the same term are assigned the same NIL ID.

Attributes: W, $\mathbf{m_l}$ As above, except we cluster NIL queries if they have the same longest mention.

Attributes: W, KB, t We take advantage of linking to Wikipedia—queries that link to Wikipedia entities outside TAC are assigned their own NIL ID. For example, `Abbas al-Musawi` is in Wikipedia, but not in the TAC KB, so if a system linked query EL1 and EL3 to the candidate for `Abbas al-Musawi`, we would assign them the same NIL ID. For cases where we had no candidates, we back off to term clustering as above.

Attributes: W, KB, $\mathbf{m_l}$ As with KB or t, except that we back off to the longest mention.

Context clustering In effort to move beyond rule-based systems, our other clustering method uses the *context* of each query chain. The clustering is implemented with the hierarchical clustering package from SciPy[17] using `single` linkage method and `cosine` metric. All queries are clustered using the following features: untokenized query name and unigram (case-normalised and without stopwords) counts from sentences containing NEs from the query's coreference chain. Clusters are flattened using the `distance` threshold of 0.5 and if a cluster contains a Wikipedia title mappable to the TAC KB, that is chosen as the final ID, otherwise a NIL ID is generated. For example, if two queries named "Tom Cruise" cluster together and the first had been linked to the TAC KB entry for `Tom Cruise` and the second to NIL, both will inherit the appropriate entity ID. Entity ID disagreements are resolved by choosing one at random.

This concludes our description of the components we developed during the three years that we entered the TAC shared tasks. In the following section, we describe the concrete system configurations and their performance.

5.2 Performance in TAC competition

The TAC shared task has evolved during the period in which we have participated. This includes guideline changes as described above, as well as updated evaluation metrics. These are discussed in detail in Chapter 3, but we briefly review them below.

[17] http://scipy.org

The principal evaluation metric is accuracy—the proportion of correctly linked queries. A KB query is considered correctly linked if the top candidate's entity ID matches the gold standard and a NIL query is correct if identified as NIL. Partitioning of the dataset allow accuracies for only KB queries and only NIL queries, giving some indication of system performance at a finer-grain. The introduction of the NIL clustering task in 2011 added an adapted version of the B^{3+} Coreference Resolution metric (Bagga and Baldwin, 1998a) with an extra constraint that KB clusters should link to the *right* KB entry as well as being clustered together. As with the accuracy metrics, these can be reported considering only gold-standard KB and NIL clusters.

Table 5.9 shows our system results in the TAC shared task. Our systems have been competitive each year we have participated, consistently above median scores and close to the top system. In the sections below, we report official scores and analysis and compare our systems with the top system and median score, deferring detailed analysis until Section 5.3.

5.2.1 2010

Our three 2010 submissions are based on our reranking pipeline, using the 2009 Wikipedia snapshot and the same NER and in-document coreference. The baseline (10.2)[18] uses tuned search followed by a cosine similarity disambiguator, scoring the candidate by `unigram_cosine`. Another (10.3) uses the alias index, followed by our Cucerzan (2007) reimplementation with article graph reranking. The full reranking pipeline (10.1) adds an alias reliability filtering step to 10.3 and is our final submission.

The full reranking pipeline scores the highest with 81.9% accuracy. This is reasonably competitive with the highest scoring system at 86.8% accuracy (Lehmann et al., 2010), although their supervised system used web access during the evaluation period to query the Google Search API as a search phase, retaining the top three candidates for disambiguation. Their highest system without web access is a heuris-

[18] Our submission indexing in 2010 reflects the order in which experiments finished, rather than a principled assessment of complexity.

Submission	Accuracy			B^{3+} F		
	All	KB	NIL	All	KB	NIL
10.1: 10.3 + reliability filtering	80.9	69.0	90.8	-	-	-
10.2: Tuned search + cosine	77.7	61.0	**91.6**	-	-	-
10.3: Alias index + Cucerzan + article	**81.9**	**73.7**	88.7	-	-	-
10.1: updated	84.4	79.0	88.8	-	-	-
10.2: updated	78.5	61.1	92.9	-	-	-
10.3: updated	84.3	78.4	89.1	-	-	-
Median	68.4	-	-	-	-	-
Max: Lehmann et al. (2010)	85.8	79.2	91.2	-	-	-
11.1: 10.3 + W or t	77.9	-	-	75.3	**65.5**	**85.1**
11.2: 10.3 + W or m_l	77.9	-	-	75.3	**65.5**	85.0
11.3: 10.3 + W or KB or t	77.9	-	-	**75.4**	**65.5**	**85.2**
Median	-	-	-	71.6	-	-
Max: Monahan et al. (2011)	86.1	-	-	84.6	76.2	93.0
12.1: Unsupervised + W or KB or t	72.2	-	-	**66.5**	**65.6**	**67.5**
12.2: Supervised + W or KB or t	68.0	-	-	61.0	56.8	65.6
12.3: Unsupervised + context	72.2	-	-	58.8	**65.6**	49.1
12.4: Supervised + context	68.0	-	-	54.0	56.8	48.9
Median	-	-	-	53.6	49.6	59.4
Max: Cucerzan (2012)	76.6	-	-	73.0	68.5	78.1

Table 5.9: Overview of TAC results in 2010, 2011 and 2012. Bold results indicate the figure for a particular metric of our systems.

tic combination of features and scores 85.8% accuracy, around 4% higher than 10.3. Bugs in our implementation discovered while writing the system description paper meant that we could report higher scores: 84.4% accuracy for 10.1 is ranked second to Lehmann et al., but is unfortunately not an official figure.

Table 5.10 shows accuracies by query entity type and document genre. In general, PER queries are the easiest to link, followed by ORG and GPE. No single system performs better on all entity types, although the cosine baseline, 10.2, is markedly worse on GPE

Submission	All			News			Web		
	PER	ORG	GPE	PER	ORG	GPE	PER	ORG	GPE
10.1: Pipeline + reliability	94	74	**74**	82	73	75	87	77	**74**
10.2: Cosine	92	**78**	63	96	**74**	61	84	**86**	67
10.3: Pipeline	**95**	77	74	**98**	73	**75**	**90**	84	71

Table 5.10: Accuracies on TAC 10-eval by genre and entity type.

queries, contributing to its lower performance.[19] We do not find any major difference in overall performance across genres, but ORG queries are linked more accurately in web documents than PER queries. This is somewhat surprising as web documents are typically noisier and less well-edited than newswire.

5.2.2 2011

Our 2011 submissions reuse the full pipeline from our TAC 10 system (i.e. 10.3) and added different clustering rules: term (11.1), longest mention (11.2), KB or term (11.3). These naïve rule submissions perform similarly at 75.3 and 75.4 B^{3+} F all. Again, this is between the median and top score, but closer to former.

The top system by Monahan et al. is an extension of Lehmann et al. (2010) using an *inductive* approach. All queries are linking then passed to a four stage clustering process. Clustering all queries means that the system can recover from incorrect linking decisions and potentially change the target of difficult queries that cluster with less ambiguous queries. Pairs of normalised query terms are assigned a distance by a supervised logistic regression model, which is used to agglomeratively cluster the mentions. The features include entity type, assigned links, term similarity and local context features. These are the noun phrases that contain the query term, often equivalent to a containing entity mention. The first phase clusters are merged using a combination of the linked mention and local context features. The final entity ID is assigned to each cluster, based on the majority vote of its query members. Monahan

[19] The entity type distribution is balanced in the 10-eval dataset.

Sub.	All			News			Web		
	PER	ORG	GPE	PER	ORG	GPE	PER	ORG	GPE
11.1	76	**73**	**77**	80	73	**74**	69	73	**82**
11.2	76	73	77	80	73	**74**	70	**73**	82
11.3	**77**	**73**	76	**80**	**73**	74	**70**	73	82

Table 5.11: B^{3+} F scores on TAC 11 by genre and entity type.

Entity type	B^{3+} F	
	KB	NIL
PER	23.7	94.0
ORG	44.6	87.1

Table 5.12: Newswire B^{3+} F scores for 11.3 by query type and entity type.

et al. report linking accuracy and clustering scores that are substantially higher than 11.3: 8.2% and 9.2% respectively.

Table 5.11 shows an analysis by genre and entity type. As seen in the previous year, PER scores are higher in newswire documents, but GPE queries are easier to cluster in web documents. NIL and KB clustering scores are balanced for most combinations of genre and entity type, except for newswire PER and ORG entities. Table 5.12 shows that PER and ORG KB queries in newswire are clustered far worse than their NIL equivalents by system 11.3. A partial explanation is that once filtered for entity type and genre, KB queries make up the minority of the remainder; only 99 of 500 PER newswire queries are in the KB. This is the least balanced subset (w.r.t. query type) of the data and so perhaps the system tends to mis-link more NIL queries to a KB entry.

While less accurate, our systems show that simple clustering approaches can be effective, although either more sophisticated linking or clustering is required to keep pace with the gains made by other teams.

5.2.3 2012

Our 2012 systems take a substantially different approach to linking and are based on the unsupervised and supervised system as described in subsection 5.1.4, combining features by averaging or a linear regression model. These are combined with the KB or t rule clustering and context clustering. The combination of linker and clustering produces four submissions: unsupervised with rules (12.1) and with context (12.3), and supervised with rules (12.2) and with context (12.4).

The top scoring system (Cucerzan, 2012, 2011) is an adaption of a production system that extends Cucerzan (2007), adding a supervised linear model, non-entity topic features, geolocation features and the ability to postpone final mention boundary detection—an extraction step—until disambiguation, making it more robust to pipeline error. In contrast to other approaches, no explicit clustering step is used and the system uses high quality linking and mapping from the larger Wikipedia to the TAC KB to assign queries to NIL clusters.

Our best system 12.1, at 66.5% B^{3+} F, performs moderately well relative to the top and median scores. However, our KB B^{3+} F score at 65.6% is more competitive, only 3.1% from the top score, reflecting the higher priority we place on improving linking to the KB over clustering. Our NIL context clustering scores are substantially below median where the naïve clustering scores are above, suggesting these coarser methods (without distance parameters, etc.) are more robust to any dataset variation, as context clustering was comparable in development experiments on TAC 11. Our supervised system's poor performance is frustratingly below that of the unsupervised system where the literature suggest otherwise (Ji et al., 2011).

Table 5.13 shows results on different genres and entity types. These follow a similar pattern to those above—linking PER queries and newswire text is easier to link, but ORG queries are easiest to link in web text. Our submissions in 2012 concentrate on improving linking accuracy and experimenting with more sophisticated clustering.

Sys.	All			News			Web		
	PER	ORG	GPE	PER	ORG	GPE	PER	ORG	GPE
12.1	**74**	**60**	**63**	**75**	**61**	**69**	**69**	**58**	**53**
12.2	71	56	51	72	59	57	69	51	42
12.3	60	53	62	61	53	68	55	52	52
12.4	59	49	51	60	52	56	56	46	41

Table 5.13: B^{3+} F score on TAC 12 by genre and entity type.

Our supervised system performs poorly, but our unsupervised system with rule-based clustering is surprisingly competitive.

5.3 A state-of-the-art linking system

We now present current systems that are extensions of those submitted to TAC in 2010, 2011 and 2012: an unsupervised ($\mathcal{U}$) and two supervised systems ($\mathcal{S}$, $\mathcal{S}+$).

Our unsupervised model features and settings are as in TAC 12, but we add a score threshold to improve NIL query accuracy. If the maximum score of a mention's candidates is lower than a threshold, we link the mention to NIL. We optimised a threshold of 0.2 for the unsupervised system using the TAC 09 data, and we use this for all experiments.[20] Using a threshold inevitably mis-classifies some KB queries as NIL, but optimising over accuracy balances this effect.

Our supervised systems use a linear regression model with L2 regularization implemented using SCIKIT-LEARN (Pedregosa et al., 2011). To train these models, we first link with the unsupervised system, then select the top three candidates as training instances (rather than the five we used before). The Wikipedia mapping strategy has subtle implications on training. A NIL query does not have a valid candidate in the TAC KB, but may have one in Wikipedia, as it may have been added since or may have had no infoboxes when the KB is constructed—we refer to these as TAC NILs. Care must be taken to exclude TAC NIL candidates during training, as the correct candidate for a TAC NIL will have a label that is inconsistent with its features. For example, Example 2 should link to NIL, as there is no entry for Abbas Moussawi in the TAC KB. The correct entry, `Abbas al-Musawi`, is found in our Wikipedia snapshot and our system may link the query to it. Despite being correct, this will be marked incorrect for training, which is inconsistent training data for the model. This was the cause of our poor performing supervised system in TAC 12. The base supervised system ($\mathcal{S}$) trains on queries from the TAC 09, TAC 10 training and TAC 10 evaluation datasets. The extended system ($\mathcal{S}+$) trains on the same data, but also TAC 11, using all available data for linking TAC 12 queries. We use some of our previously defined features and also the following new features:

[20] Except on TAC 09, where we use a threshold of 0, as this was the tuning dataset.

Unsupervised score The unsupervised system has been a strong baseline system in
unsup_score our TAC submissions and so we use the score as a feature in our supervised model to summarise those features.

Candidate set statistics Our previous features model how well a mention matches a KB entry. These do not model NIL queries well where, rather than positive linking evidence, systems should identify *lack* of evidence or a reasonable candidate. We calculate features that attempt to summarise the set of candidates and a particular
cand_stats candidate's place within it. After other features have been calculated, we calculate the minimum, maximum, mean and entropy of each feature across the candidate set and add these values as features. These features model candidate sets with low mean feature scores, which we might expect to occur for NIL queries, where there is no good match.

We also calculate the inverse rank of each candidate with respect to each feature. For example, if a candidate has the highest `unsup_score` in a list of ten candidates, it
top_prop would have a `irank_unsup_score` of 0.1 ($\frac{1}{10}$). We also count the proportion of features for which a candidate has the highest values. The motivation for these features is to capture a coarse-grained feature combination.

Tables 5.14 and 5.15 show the configurations of our state-of-the-art systems: $\mathcal{U}$, $\mathcal{S}$ and $\mathcal{S}+$. These include some features from our TAC submissions and not all our historical features in Table 5.2 are included.

Phase	Components
Resources	Wikipedia 04/2012, crosswikis, alias generation
Extraction	NER, in-document coreference
Search	Alias index, query expansion
Disambiguation	Unsupervised reranking with features: `alias_cosine`, `cat_score`, `context_score`, `kb_prior`, `kb_refprob`, `sent_cont`, `top_overlap`
NIL detection	Threshold: 0.2 (except for TAC 09)
NIL clustering	Rule clustering: W or KB or m_l

Table 5.14: Configuration for the unsupervised NEL system ($\mathcal{U}$).

Phase	Components
Resources	Wikipedia 04/2012, crosswikis, alias generation
Extraction	NER, in-document coreference
Search	Alias index, query expansion
Disambiguation	Unsupervised reranking as per $\mathcal{U}$. Supervised reranking of top-3 candidates with features: `alias_cosine`, `cat_score`, `context_score`, `kb_prior`, `kb_refprob`, `sent_cont`, `top_overlap`, `ttl_cosine`, `ttl_edit`, `ttl_dice`, `ttl_jw`, `chain_kb_refprob`, `unigram_cosine`, `bigram_cosine`, `sent_cont`, `type_match`, `type_combo`, `type_match_l`, `per_match`, `loc_prep`, `unsup_score`, `cand_stats`, `top_prop`. $\mathcal{S}$ is trained on TAC 09, TAC 10 training, TAC 10 data, $\mathcal{S}+$ includes TAC 11 data.
NIL detection	Threshold: 0.2 (except for TAC 09)
NIL clustering	Rule clustering: W or KB or m_l

Table 5.15: Configuration for the supervised NEL systems ($\mathcal{S}$ and $\mathcal{S}+$).

System	Dataset	Accuracy			B^{3+} F		
		All	KB	NIL	All	KB	NIL
$\mathcal{U}$	09	81.8	77.1	85.3	-	-	-
Best TAC, Varma et al. (2009)	09	82.2	86.5	76.4	-	-	-
Han and Sun (2011)	09	86.0	79.0	90.0	-	-	-
$\mathcal{U}$	10	84.8	79.8	88.9	-	-	-
Best TAC, Lehmann et al. (2010)	10	86.8	80.6	92.0	-	-	-
Cucerzan (2011)	10	90.0	87.3	92.2	-	-	-
$\mathcal{U}$	11	87.2	81.2	93.1	84.1	79.5	88.8
$\mathcal{S}$	11	89.2	83.1	95.4	86.3	81.3	91.2
Best TAC, Monahan et al. (2011)	11	86.1	-	-	84.6	76.2	93.0
Zhang et al. (2012)	11	87.6	-	-	-	-	-
$\mathcal{U}$	12	74.4	70.8	78.4	70.1	67.2	73.4
$\mathcal{S}$	12	74.7	66.2	84.2	70.4	62.9	78.9
$\mathcal{S}+$	12	75.8	69.1	83.3	71.5	65.7	78.0
Best TAC, Cucerzan (2012)	12	76.6	-	-	73.0	68.5	78.1

Table 5.16: Results on TAC datasets with current linkers.

5.3.1 Results on TAC datasets

We evaluate our systems using TAC data, presenting analysis to given some insight into the effect of thresholds, the impact of different features and an exploration of different error types.

Table 5.16 summarises the performance of our systems compared against the top TAC submissions and other reported results. Since we train our supervised systems on TAC 09 and TAC 10 data, we only report results of our unsupervised systems for those years. These show similar accuracies to the top-ranked TAC systems in those years: 0.1% higher in 2009 and 0.9% lower in 2010, but are below the best systems reported since in the literature. Both unsupervised and supervised systems perform better than Monahan et al.'s system in terms of accuracy, and high quality linking means that the naïve rules induce good clusters. Finally, the supervised system performs well on

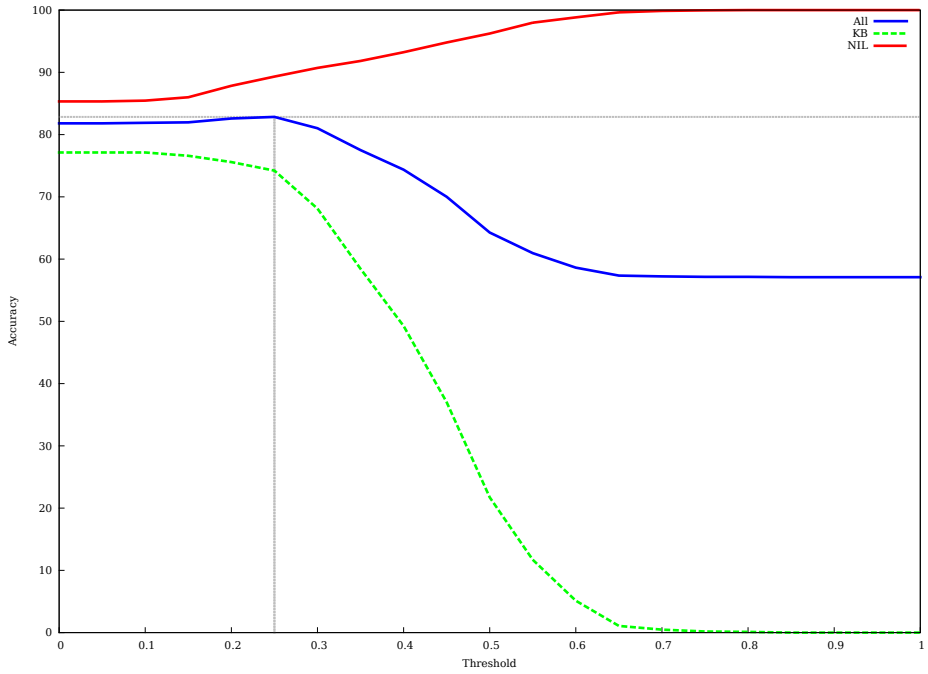


Figure 5.4: Ideal NIL threshold for the unsupervised NEL system ($\mathcal{U}$) on TAC 09

TAC 12, especially when trained on all available data ($\mathcal{S}+$), a substantial improvement over the unsupervised system's performance. Even so, the supervised system in Cucerzan (2012) performs 0.4% higher in accuracy, 1.1% better for B^{3+} F all.

5.3.2 Tuning the NIL threshold

Thresholds are a naïve approach to NIL classification. The ideal threshold depends on the distribution of KB and NIL queries amongst the dataset and the linker's performance. We optimised a threshold on TAC 09 and used that for all linkers on later datasets. Figure 5.4 shows the relationship between overall accuracy (blue, solid), KB accuracy (green, dashed) and NIL accuracy (red, dotted) as we vary the threshold value. A threshold of 1.0 would classify every query as NIL and result in 100% NIL accuracy, but 0% KB accuracy due to failure to recall any KB queries. Conversely, no threshold means any NIL query with at least one candidate will be classified incorrectly. An ideal threshold trades these off and this is indicated by a dotted line showing the maximum accuracy and the threshold that produces it. For the TAC 09 data, the relationship between KB and NIL accuracies is stable until around 0.3, when

System	Data	Best threshold	Best Accuracy	Accuracy @ 0.2	δ
$\mathcal{U}$	09	0.25	82.8	82.6	-0.3
$\mathcal{U}$	10	0.00	85.9	84.8	-1.1
$\mathcal{U}$	11	0.15	88.0	87.2	-0.8
$\mathcal{S}$	11	0.10	90.0	89.2	-0.8
$\mathcal{U}$	12	0.25	75.5	74.3	-1.1
$\mathcal{S}$	12	0.15	75.0	74.7	-0.4
$\mathcal{S}+$	12	0.15	75.9	75.8	-0.0

Table 5.17: Threshold analysis. We show the ideal threshold and accuracy for different systems on datasets, including the accuracy and delta at 0.2.

the two accuracies diverge as the NIL accuracy increases at the expense of the KB accuracy. We optimize in increments of 0.05 and sometimes this produces a range of thresholds with the same accuracy.

Choosing a threshold of 0.2, the optimum for TAC 09, is reasonable. The TAC 09 data has more NIL queries than KB and thresholds that benefit NILs (i.e. high) will have a greater effect on overall accuracy. Low thresholds will benefit datasets with a more even distribution of NIL and KB queries or those skewed to a high KB proportion.

Table 5.17 shows the optimal accuracies and thresholds for our systems over different datasets. The effect of the 0.2 threshold is illustrated, showing accuracy and the difference from the optimal accuracy figure. This difference is greater on datasets with a higher KB proportion, but less on more balanced datasets. The range of ideal threshold shows the impact of the different query type distributions and no one threshold is optimal for all datasets.

5.3.3 Feature analysis

Our two systems use different features to model linking. This subsection presents an ablation study to compare their relative importance by omitting each feature in

Linker	09	10	11	12
$\mathcal{U}$	81.8	84.8	87.2	74.3
-alias_cosine	+0.1	-5.5	-7.0	-3.3
-cat_score	+0.1	+0.1	+0.4	-1.3
-context_score	0.0	-0.2	+0.3	-1.0
-kb_prior	0.0	+0.1	+0.4	-1.3
-kb_refprob	-3.3	-4.7	-4.5	-4.5
-sent_cont	+0.2	-0.3	0.0	-0.4
-top_overlap	-3.1	-2.1	-6.4	-11.5

Table 5.18: Feature ablation using the unsupervised NEL system ($\mathcal{U}$).

turn and evaluating. Table 5.18 show the accuracy of the full unsupervised model for various datasets, then accuracies without each feature. A large negative difference shows that the model depends heavily on that feature.

On average, the most influential features are `top_overlap`, `alias_cosine` and `kb_refprob`, which all result in at least 4% lower accuracy when omitted. Each of these features depend heavily on the KB structure. The most influential, `top_overlap`, uses the Wikipedia article graph, where `alias_cosine` and `kb_refprob` use alias information from Wikipedia redirects and hyperlink anchors. The remaining features are far less important at 0.3% to 0.4% loss. Again, we see a wider variance between the datasets. For example, the `alias_cosine` feature represents close name matches. Surprisingly, this benefits the linker substantially in all datasets but TAC 09, where removing it increases accuracy.

Table 5.19 shows the accuracies for the supervised system, $\mathcal{S}$, without each feature. The `unigram_cosine` feature has a consistently high impact on the supervised models, showing that document-entry similarity is central to our supervised system. Interestingly, we tried adding this to the unsupervised system, but it degraded performance. The next highest impact features are `cat_score`, `kb_score`, `cand_stats` and `kb_refprob`. The `cat_score` is also included in the unsupervised model, but did not have a great impact when removed, and `kb_score` summarises the candidate's

Linker	11	12	Linker	11	12
$\mathcal{S}$	89.2	74.7	$\mathcal{S}$	89.2	74.7
-alias_cosine	+0.4	+0.1	-bigram_cosine	0.0	-0.4
-cand_stats	-2.8	-3.5	-cat_score	-2.8	-3.1
-chain_kb_refprob	-0.1	-2.1	-context_score	-0.7	-2.0
-kb_prior	+0.8	+0.1	-kb_refprob	-2.8	-3.2
-kb_score	-5.5	-2.6	-loc_prep	+0.4	-0.4
-per_match	+0.5	-0.1	-sent_cont	-0.8	-1.5
-top_overlap	-1.0	-2.7	-ttl_contains	+0.4	0.0
-ttl_cosine	+0.3	-0.2	-ttl_dice	+0.6	-0.4
-ttl_edit	+0.3	-0.2	-ttl_ends	+0.4	-0.1
-ttl_jw	+0.7	+0.2	-ttl_starts	+0.6	+0.2
-type_combo	0.0	0.0	-type_match	+0.6	+0.3
-unigram_cosine	-5.3	-4.5	-unsup_score	-0.9	-1.2

Table 5.19: Feature ablation using the supervised NEL system ($\mathcal{S}$).

alias match. Modelling the other candidates is also useful for linking, as `cand_stats` and `top_prop` have a moderate impact. The unsupervised system depends on the `kb_refprob` and `alias_cosine` features, but the latter plays a minor role in the supervised model. We see less inter-dataset variance than for the unsupervised system.

5.3.4 Error distribution

In general, shared tasks must provide a common evaluation to often diverse systems and TAC is no different. Accuracy and B^{3+} F summarise overall performance well for comparative purposes, but give limited insight into the behaviour of complex systems. We define a categorisation of error types that quantitatively illustrate what kind of errors a system makes.

Recall This is the proportion of queries that are not linkable. A KB query is *linkable* if its entity is included in the candidate list. As we do not know the correct entity

for a NIL query, they are all deemed linkable. This is a useful measure of search effectiveness as KB queries are only linkable if their entity is retrieved.

Wrong KB This error results when a KB query is linked to the wrong KB entry.

KB to TAC NIL This error, specific to the Wikipedia mapping approach to the task, occurs when a KB query is linked to a Wikipedia entity that does not exist in the TAC KB. This is essentially a Wrong KB error that is evaluated as a NIL.

KB to NIL This error is caused by linking a KB query to NIL, possibly by failing to retrieve the candidate or by somehow classifying it as NIL (in our case when the score falls below the NIL threshold).

NIL to KB The final error results from linking a NIL query to a KB entry—the inverse of KB to NIL.

Table 5.20 show the distribution of error types in the systems. The recall error is around 5% or lower, showing that most queries are linkable, with at worst 90% recall on KB queries given both datasets are balanced. Incorrect mention boundaries or oblique references are possible explanations for these errors, but they are relatively rare. Errors where a systems selects the wrong KB entity account for between 30% and 50% of their errors (Wrong KB + KB to TAC NIL), but on average around 40% of these errors are eventually classed NIL by the accuracy metric as they are outside the TAC KB. So where systems use the Wikipedia mapping approach, their mis-linking errors can be obscured when measuring accuracy. The remaining error classes, KB to NIL and NIL to KB, show the effect of NIL classification and typically account for the majority of errors.[21] While confusing the entity that a query should be linked to is a problem, recognising NIL queries is the main challenge our systems face.

5.3.5 End-to-end linking

All evaluation so far has been on the TAC datasets, a query-based understanding of the linking task. Using queries factors out mention detection and fixes the number of

[21] Recall here that we optimise our threshold on TAC 09 and so our system does not use *any* method to classify them, leading to a high proportion of NIL to KB errors.

Error	$\mathcal{U}$ 09	$\mathcal{U}$ 10e	$\mathcal{S}$ 11	$\mathcal{U}$ 11	$\mathcal{S}$ 12	$\mathcal{U}$ 12
Recall	4.6	2.7	2.6	2.6	5.2	5.2
Wrong KB	194	64	48	65	151	161
KB to TAC NIL	158	64	25	38	56	77
KB to NIL	31	78	117	108	191	106
NIL to KB	327	136	52	78	166	227
TOTAL	710	342	242	289	564	571

Table 5.20: Error profiles on TAC datasets with current linkers

System	All	KB	NIL
$\mathcal{U}$	70.4	72.0	65.6
$\mathcal{S}$	69.3	72.4	61.5
$\mathcal{S}+$	69.0	72.5	60.4
$\mathcal{S}$SMH	71.7	75.1	63.3

Table 5.21: Performance on the SMH DEV dataset.

mentions to be linked per document.[22] While this simplifies evaluation, some tasks may require linking all mentions in the document.

We evaluate our systems on the SMH dataset introduced in Chapter 3, a corpus of Australian news stories. The task is harder in this end-to-end setting. Our annotations specify gold-standard mentions and links to Wikipedia or NIL. We consider a mention correct if we retrieve the exact bound and link to the correct title or NIL. This is extremely sensitive to pipeline error as the system cannot recover from NER errors. We calculate set-based precision, recall and F for all mentions and the KB and NIL subsets. The evaluation is not perfect, as we do not penalise the system for producing mentions outside the gold-standard, but these should mostly correspond to NER errors, which are penalised as we cannot link them.

Table 5.21 shows the performance of our systems on the DEV section of the SMH corpus. The unsupervised system performs better than the supervised system trained

[22]This is with respect to the evaluation, the same document can be used in multiple queries.

System	All	KB	NIL
$\mathcal{U}$ (t=0.25)	71.3	72.9	66.7
$\mathcal{S}$ (t=0.1)	70.6	72.1	65.8
$\mathcal{S}+$ (t=0.1)	70.7	72.3	65.9
$\mathcal{S}$SMH (t=0.2)	71.7	74.2	65.1

Table 5.22: Performance on the SMH TEST dataset. Thresholds optimised on SMH DEV.

on TAC, 0.9% better at 70.4%, and the supervised system with more TAC data, $\mathcal{S}+$, performs worse again. The relative success of the unsupervised system is somewhat surprising and may indicate that our models learned on TAC queries do not generalise well. TAC queries target ambiguous mentions and perhaps our models are too specific to effectively link all kinds of mentions, so a model that uses heuristics has more robust performance. Retraining the supervised system on the SMH TRAIN section improves performance to 71.7%. We use the threshold of 0.2 optimised on TAC 09 for all systems. Note that we do not change the training process and only consider candidates to be labelled LINK if they appear in the TAC KB. While this performance is reasonable, we might expect better given that newswire data in TAC is usually linked fairly well. Given that the main difference is that we extract and link *all* mentions, we measured the upper bound after the extraction and search. The proportion of mentions that are "linkable" is 83.45%, substantially lower than the 97.4% we find in TAC 11 and 94.8% in TAC 12. This is a significant barrier to end-to-end linking performance, but is not often considered in linking evaluations, which either use a query-based TAC dataset or use gold-standard mentions.

One aspect of adapting a TAC linker to other tasks is that we might retrain on labelled data from the new task, as we have done above. The other is that the NIL parameter can be tuned. We do so here on the DEV data and evaluate over the TEST split. Table 5.22 shows the same performance difference between TAC unsupervised and supervised systems: the unsupervised system is more robust to changed data.

Retraining on the SMH TRAIN split shows the best performance, but only 0.4% above the unsupervised system once thresholds have been optimised.

5.4 Discussion

Shared tasks have had a major impact on NLP research by concentrating research effort on a common evaluation and data. Despite these benefits, the competitive nature of shared tasks can encourage solutions that pursue performance as the only goal. The challenge in long-running tasks is to continue developing task definitions and evaluation in response to these problems. This is certainly the case in the TAC task.

Linking accuracy is the initial evaluation metric in the 2009 and 2010 tasks and the top systems report 82.2% and 86.8% (Varma et al., 2009; Lehmann et al., 2010). This is a good assessment of KB linking, but the treatment of NILs only extends to identification and presents a limited evaluation; NILs are somewhat a second-class citizen. The TAC 11 task introduces NIL clustering.[23] This requires a more sophisticated treatment of NILs and with lower top scores of 84.6% B^{3+} F in 2011 and 73% B^{3+} F in 2012. While the B^{3+} metrics better evaluate clustering, they have a less straightforward interpretation than accuracy.

The dataset can have a large effect on the difficulty of the task and interact with the evaluation metric in interesting ways. The TAC organisers selected intentionally "confusable entities" for their dataset, preferring incorrect spelling, abbreviations and ambiguous names (Ji et al., 2010). This, coupled with the individual query granularity of the task means that it is unclear how linking TAC queries relates to linking entire documents. As we reported in Chapter 3, the datasets vary in ratio of NIL to KB queries and mention to entity ambiguity. Systems optimised on a particular dataset may not perform well on another dataset with a different distribution.

Like many systems, we link to Wikipedia then map to the TAC KB. The advantage is linking against a larger, richer dataset, but optimises for the task rather than the un-

[23] Early guidelines discussed the NIL clustering task.

derlying real-world problem. Wikipedia editor guidelines and behaviour change over time and complexity of the markup scheme and size of the dataset make processing time-consuming and difficult.

Linking to a larger KB risks having greater numbers of ambiguous entities, (i.e. more `John Smith`s), or entities that are less distinct, such as organisations and their subsidiaries. This means that approaches need even more precise modelling of entity context for linking. As the KB evolves, entries may be split or merged between versions and titles may be changed. We describe our strategy for handling this above, but this requires a simple mapping between KBs, such as shared titles. Other KBs may require linking to Wikipedia before linking documents can be attempted.

The temporal distance between KB and documents for linking also impacts the task. The original TAC source documents and KB are from the same era: 2007–2008, while many systems effectively link against future KBs (i.e. from 2012). This situation is realistic for some retrospective use-cases—some documents predate the KB—but does not represent the prospective use-case where the KB is updated with entities from new stories. Specifically, we can classify NIL mentions into two classes: emerging and non-notable.[24] The former may be NIL at the time of document publication, but their future notability will ensure inclusion in a later version of the KB. The other class of NILs never pass the notability threshold of the KB. So, for emerging NILs, using a later KB is tantamount to peering into the future to link them. This will have particular impact on context similarity features. A future entry for an entity may contain references to the same event that the document is reporting on for the first time. The strong match between entry and document would not occur in the prospective case. Linking to a dynamic KB is a key unsolved problem.

Replication of experimental results is a key challenge for any scientific field. There have been several responses within computational linguistics: analyses (Fokkens et al., 2013), interpretations of performance differences (Berg-Kirkpatrick et al., 2012) and calls-to-arms (Pedersen, 2008). The shared task is, by design, inclusive, but peer-

[24]The notability criterion is specific to Wikipedia.

review of system description papers can run counter to this—it is not inclusive to reject a paper—or not the most efficient use of limited resources. The TAC papers do not undergo a standard peer review, although "non-responsive" papers can be excluded[25] and authors are encouraged to submit work to other venues. As such, there is often less scrutiny applied to a TAC shared-task paper than to one submitted to conference or journal reviewers. Without effort from shared task organisers and authors, this can make it difficult to replicate systems and their results. The TAC organisers have taken steps to track year-on-year performance (Ji et al., 2011), but the level of detail in individual system papers varies substantially.

Indeed, any system using large, noisy resources such as Wikipedia may require too many procedures for handling edge cases to effectively document, something only exacerbated by the competitive mechanism that makes shared tasks so successful. Releasing systems would help replication, but may be a barrier to entry for some commercial teams. The top systems in TAC 10, 11 and 12 are all the product of commercial research and a public release seems unlikely.

We have critiqued some of the issues concerning evaluation, Wikipedia mapping and replication. Despite these issues, the TAC shared task is a productive and vital part of research into NEL.

5.5 Summary

This chapter reports on three years of participation in the TAC named entity linking shared task. We describe the task and principles that guide our approaches. We detail the resources and components that make up our three submitted systems and compare their performance with other competitors. Finally, we present two systems that build on the experience of TAC participation and conduct a detailed analysis, concluding with some discussion of the shared task experience.

[25] www.nist.gov/tac/2013/KBP/reporting_guidelines.html

We review again our design goals to underline their contribution. Linking the whole document is crucial, as information from one mention can help link another, as relationships between entities in the KB are often mirrored in text. NEL systems are complex and require careful engineering to create a system that is efficient enough to support large research programmes and flexible enough that it can be adapted to suit new research directions. While not all NEL systems are implemented as strict pipelines, most are, so tools that allow evaluation at each step are invaluable in minimising cascading errors. NEL is a knowledge-intensive task, and we have found using the Wikipedia mapping technique very useful, but recognise that this has implications. Finally, our system shares attributes of many others, but we put particular emphasis on recovering linguistic detail to help disambiguate mentions, for example resolving in-document coreference.

Although our systems show state-of-the-art performance, linking is far from a solved problem. In the following chapters, we explore a deeper analysis of the language with which writers describe entities and how this can help disambiguation.

6 Extracting Apposition

> The evening is also expected to be the first public outing for {another pair of lovebirds}$_h$: {Symond senior and {{his new girlfriend}$_h$ {Amber Keating}$_a$}$_h$, {the ex-wife of {Patrick Keating}$_h$, {the son of {the former Labor prime minister}$_a$ {Paul Keating}$_h$}$_a$}$_a$}$_a$.
>
> Nested apposition in Hornery, SMH 2012-02-15[1]

We have focused on automatic methods for disambiguating entities, but it is also a task that human language users face: a reader must disambiguate entities using cues supplied by the writer. Apposition is a linguistic structure that provides additional information about a concept and can be used to disambiguate it. Example 1 shows a simple example of how the author can use apposition to mention a named entity ({Tony Abbott}$_h$) and specify information about his professional role ({the leader of the opposition}$_a$). The current role is incidental and not the focus of the sentence, which is that a change is imminent.

(1) {Tony Abbott}$_h$, {the leader of the opposition}$_a$, can expect to change his CV tomorrow.

This chapter explores apposition, how it is defined and previous work on extraction. We contribute an analysis of apposition annotation in the OntoNotes 4 dataset. We present several techniques for extracting apposition from text that improve on the

[1] www.smh.com.au/lifestyle/fill-im-up-lunch-is-on-them

state of the art. More specifically, these contributions are: syntactic restrictions that better model the linguistic theory of apposition, semantic feature that encode *what* information the apposition introduces, and a joint model of apposition. Preliminary extraction results are presented in Radford and Curran (2013).

6.1 Apposition

We define apposition fairly restrictively as a structure composed of two or more adjacent coreferent noun phrases (NP). The earlier example is fairly straightforward and explicitly marked using commas. Example 2 shows a more complex example, consisting of three comma-separated NPs—the first NP (HEAD) names an entity and the others (ATTRs) supply age and profession attributes.

(2) {John Ake}$_{\text{h}}$, {48}$_{\text{a}}$, {a former vice-president in charge of legal compliance at American Capital Management & Research Inc., in Houston,}$_{\text{a}}$, ...

Attributes can be difficult to identify despite characteristic punctuation cues, as punctuation plays many roles and attributes may have rich substructure, including nested apposition, such as the very complex example at the preface to this chapter.

This section surveys the various ways apposition has been defined in the literature. We then show the broad range of tasks that use extracted appositions and conclude with a deeper analysis of work that intrinsically evaluates apposition extraction (Favre and Hakkani-Tür, 2009).

6.1.1 Defining apposition

Apposition is widely studied, but "grammarians vary in the freedom with which they apply the term 'apposition' " (Quirk et al., 1985). We refer readers to Quirk et al. and Meyer (1992) for extensive studies of apposition and outline the key points below.

Apposition is usually composed of two or more adjacent NP, or *units*, hierarchically structured, so one is the *head* NP (HEAD) and the rest *attributes* (ATTRs). They are often flagged using punctuation in text and pauses in speech. Pragmatically, they allow an author to introduce new information and build a shared context (Meyer, 1992).

Quirk et al. propose three tests for apposition: each phrase can be omitted without affecting sentence acceptability; each fulfils the same syntactic function in the resulting sentences; and extralinguistic reference (i.e. coreference) is unchanged. One consequence of these tests is that some pairs of units that appear to be appositions are not. For example, Sydney, Australia is not an apposition despite the comma and disambiguating country information, since the units refer to different entities.[2]

Meyer (1992) expands on the "conceptually sound", but "incomplete" analysis in Quirk et al. (1985), admitting a wider range of constructions defined along syntactic, semantic and pragmatic dimensions (Meyer, 1992, p. 5). As well as detailed qualitative analysis, Meyer also contributes evidence of the distribution of different types of apposition. Table 6.1 reproduces the distribution of syntactic forms Meyer found in the Brown corpus (Kucera and Francis, 1967). The implications of the more relaxed scheme are evident; some cases, particularly NPs in apposition with clauses seem *too dissimilar* to the conventional apposition we see in Example 1.[3]

Apposition can correspond to several semantic classes depending on whether the ATTR is more (i.e. identification/appellation), less (i.e. characterisation) or equally specific (i.e. paraphrase) in reference. Meyer finds that a reference relation holds between the units in 62% of all nominal apposition, with an overall 36% of those relations coreference (the remainder are part-whole or cataphoric). Semantic constraints can help resolve ambiguity between commas used for lists and appositive commas as adjacent NPs in a list may share the same NE type. For example, world knowledge is is required to know that Syria, Libya is not an apposition where Istanbul, Constantino-

[2]We explore these structures further in Chapter 7.

[3]Meyer proposes a notion of *gradable* apposition to handle cases that seem appositional but do not fit stricter schemes.

Form and example	Count
Nominal apposition {The first twenty thousand pounds}$_h$, {the original grant}$_a$, is committed.	647
NPs in apposition with clauses or sentences There is perhaps no {value statement}$_h$ on which people would more universally agree than {the statement that intense pain is bad}$_a$.	152
Appositions with obligatory markers: NP + NP . . . {problems}$_h$ such as {those touched on in the last few paragraphs}$_a$	81
Appositions with obligatory markers: other . . . the {dorsal epithalmic}$_h$ or {habenular}$_a$ region . . .	19
Non-nominal apposition More stands on the margins of modernity {for one reason alone}$_h$ – {because he wrote Utopia}$_a$.	127
TOTAL	1,026

Table 6.1: Syntactic distribution of apposition in the Brown corpus reproduced from Meyer (1992, p. 11, Table 2.1, Column 1 and surrounding text for examples).

ple is.[4] These constraints also rule out performance errors that appear syntactically appositional, but lack a coreference relation (e.g. . . . less, uh, fewer, people. . .).

In his pragmatic analysis of apposition, Meyer proposes that "the second unit of the apposition either wholly or partially provides new information about the first unit", finding that new information is introduced in 86% of apposition (Meyer, 1992, p. 93). He states that apposition is common in press reportage where there is a "communicative need for new information to be provided" (Meyer, 1992, p. 92). Pseudo-titles are also a common feature of the reportage and Meyer considers them to be a case of apposition where the first unit adds information. A journalist cannot assume extensive shared knowledge with the reader and so must use structures such as apposition to introduce and disambiguate entities or general concepts. This may contrast with other genres where there is more opportunity for the writer to less

[4] Although deciding which is the HEAD and which is the ATTR may prove difficult.

explicitly build context such as in a long narrative, or even omit it for stylistic effect, for example to engender a sense of confusion or pace.

We adopt the OntoNotes guidelines' relatively strict interpretation: "a noun phrase that modifies an immediately-adjacent noun phrase (these may be separated by only a comma, colon, or parenthesis)" (BBN, 2004–2007). The scheme also describes the following edge cases:

- Pseudo-titles are excluded:

  *{Building supervisor}$_a$ {Smith}$_h$ ...

  {The building supervisor}$_a$ {Smith}$_h$ ...

- Ages are included, perhaps assuming implicit content:

  ?{John Ake}$_h$, {[the] 48 [year-old]}$_a$ ...

- Adverbial phrases are included if the reference is unchanged:

  {John Smith}$_h$, {formerly the president}$_a$, ...

  *{The major tech companies}$_a$, {especially Google}$_h$, ...

The scheme defines a specificity ranking to decide which of two units is the HEAD, the higher ranked phrase in the following scale:

Proper Nouns > Pronouns > Definite NPs > Indefinite NPs > NPs

This definition of apposition is more restrictive than others—certainly closer to Quirk et al. than to Meyer—but avoids some of the complications of consistently annotating the gradable relations discussed by Meyer. Ultimately, we are also constrained by the dataset available if we are to evaluate apposition extraction without an extensive annotation effort.

6.1.2 Apposition extraction as a component

Apposition extraction is not an uncommon component in many NLP tasks, but, to our knowledge, few papers explicitly evaluate performance. The prominent role of coreference relations in apposition makes apposition extraction a natural choice for inclusion in coreference resolution systems. The syntactic characteristics of apposition have informed coreference resolution features, directly modelling apposition (Soon et al., 2001; Culotta et al., 2007), and whether two mentions share a parent (Luo and Zitouni, 2005). Features that model semantic agreement between mentions (Soon et al., 2001) can also capture the attributes added by an ATTR. In their analysis, Bengtson and Roth (2008) identify apposition as an important class of features. They also find that lists of entities can be problematic for extraction for the comma-ambiguity reasons discussed in Section 6.1.1. The coreference relations implicit in apposition can also be used to constrain unsupervised coreference resolution using Markov Logic Networks (Poon and Domingos, 2008) and help identify nominal mentions for clustering in deterministic sieves (Raghunathan et al., 2010).

Srikumar et al. (2008) explore "comma resolution" for textual entailment in the Penn TreeBank. They propose four comma classes of interest: substitute, attribute, location and list. The first two classes perform appositive functions, but are only 59% of the commas annotated. Location commas separate location names and indicate an enclosing relationship, for example city then country in "London, England" and were about 10% of cases. Lists were the last type, often lists of the same type of entity, with or without serial commas. The focus of their work was to learn rules to transform parse trees for entailment, but they were able to extract relations at 80.2% F (gold parses) and 70.4% (automatic parses).

Nenkova et al. (2005) investigate the *cognitive status* of entities in discourse—whether an entity should be familiar to the reader. Their goal is to identify commonly known entities that can be simply referred to without further explanation, saving space in a generated summary. They designed features that counted how many

times appositions were used to describe a person and the total number of apposition, relative clause or copula descriptions. In feature selection, they found that the explicit apposition feature was not important, where the overall description feature is. Identifying entity familiarity fits the NEL paradigm: Wikipedia entities are notable and probably familiar, whereas NIL entities are unlikely to be familiar.

More broadly, apposition has been exploited for a range of tasks. Apposition is a key method for specifying information and Sudo et al. (2003) examine its use in information extraction patterns, while White and Rajkumar (2008) show increases in BLEU scores for generation using a more considered treatment of punctuation, including to indicate apposition. The tight syntactic and semantic definitions mean that pairs of HEAD and ATTR are compelling sources of short, same-language parallel phrases. These are used as restated phrases for recognising textual entailment (Roth and Sammons, 2007; Cabrio and Magnini, 2010) and to identify answers to questions (Moldovan et al., 2003). Finally, while apposition carries information, it does so in a complex way. Apposition structures have been unpacked to create simpler sentences for reader comprehension (Siddharthan, 2002; Candido et al., 2009) and relation extraction (Miwa et al., 2010).

6.1.3 Evaluating apposition extraction

Favre and Hakkani-Tür (2009, FHT) evaluate three extraction systems on OntoNotes 2.9 news broadcasts. They evaluate apposition extraction in the absence of punctuation—over the output of Automated Speech Recognition (ASR). The first system retrains the Berkeley parser (Petrov and Klein, 2007) on trees labelled with apposition by appending the HEAD and ATTR suffix to NPS – we refer to this as a Labelled Berkeley Parser (LBP). The second is a Conditional Random Field (Lafferty et al., 2001, CRF) labelling words using an IOB apposition scheme. Token, POS, NE and BP-label features are used, as are presence of speech pauses. The final system classifies parse tree phrases using an Adaboost classifier (Schapire and Singer, 2000) with similar features.

The LBP, IOB and phrase systems score 41.38%, 32.76% and 40.41%, and their best system uses LBP tree labels as IOB features, scoring 42.31%. Their focus on ASR output, which precludes punctuation cues, does not indicate how well the methods perform on written text. Moreover all systems use parsers or parse-label features and do not completely evaluate non-parser methods for extraction despite including baselines.

6.2 Apposition in OntoNotes 4

We use apposition-annotated documents from the English section of OntoNotes 4 (Weischedel et al., 2011). The OntoNotes DB tool[5] is used to load the raw annotations into a MySQL database, which is transformed into a task-specific format. We applied manual adjustments to apposition that do not have exactly one HEAD and one or more ATTR[6]. Some appositions are nested, and we keep only "leaf" appositions, removing the higher-level structures. We also retain OntoNotes gold-standard POS and NE tags (18 class), as well as parse trees.[7]

The CoNLL 2011 Unrestricted Coreference Shared Task defines a scheme for allocating OntoNotes 4 documents to a training, development and test set (Pradhan et al., 2011). We follow this scheme to select our TRAIN, DEV and TEST datasets. OntoNotes 4 is made up of a wide variety of sources: broadcast conversation (BC) and broadcast news (BN), magazine (MZ), newswire (NW) and web text (WB). Table 6.2 analyses our OntoNotes 4 corpus further, with counts of words, sentences and appositions (HEAD-ATTR pairs) over TRAIN, DEV and TEST. Notably, the corpus is mostly made up of newswire, which also has the highest ratio of appositions to words.

We also replicate the OntoNotes 2.9 BN data used by FHT, selecting the same sentences from OntoNotes 4 (TRAIN_F/DEV_F/TEST_F). We do not speechify our data and again we only use leaf appositions. Favre and Hakkani-Tür (2009) used a subset of the BN corpus as they needed to align textual sentences with ASR output. With their help,

[5] http://cemantix.org/software/ontonotes-db-tool.html

[6] Available at http://schwa.org/projects/resources/wiki/Apposition

[7] These trees contain NML phrases, which we also refer to as NPS throughout this paper as they are treated equivalently.

Corpus	Words	Sentences	Apposition pairs	Apposition rate
BC	209,352	14,412	450	2
BN	226,273	12,147	740	3
MZ	197,520	8,333	627	3
NW	488,935	19,240	2,542	5
WB	169,631	8,420	510	3
TOTAL	1,291,711	62,552	4,869	4

Table 6.2: Genre analysis in OntoNotes 4 showing word, sentence and apposition pair counts. Apposition rate is the number of apposition pairs per 1,000 words.

Unit	TRAIN_F	DEV_F	TEST_F	TRAIN	DEV	TEST
Sents.	9,595	976	1,098	48,762	6,894	6,896
Appos.	590	64	68	3,877	502	490

Table 6.3: Sentence and apposition distribution.

we were able to identify the 11,669 sentences in $\text{TRAIN}_\text{F}/\text{DEV}_\text{F}/\text{TEST}_\text{F}$. The OntoNotes annotation policy for providing a syntactic tree for speaker turn changes changed between versions and so sentence offsets needed adjustment. Table 6.3 shows the distribution of sentences and appositions in the two corpora. The corpus derived from OntoNotes 4 is substantially larger than the BN-based OntoNotes 2.9 corpus.

6.2.1 Analysis

Most appositions in TRAIN have one ATTR (97.4%) with few having two (2.5%) or three (0.1%). This has two implications: that approaches examining pairs of adjacent phrases will have high coverage without having to generalize to more than two units. Secondly, we can evaluate system performance using a pairwise metric, simplifying the implementation and interpretation. For example, an apposition structured "HEAD, ATTR_1, ATTR_2" would be split into two parts for analysis and evaluation: (HEAD, ATTR_1) and (HEAD, ATTR_2).

Form	#	%	Reverse form	#	%	∑%
H t A	2,109	55.9	A t H	724	19.2	75.1
A H	482	12.8	H A	205	5.4	93.3
H t t A	85	2.3	A t t H	26	0.7	96.3
H t A t A	23	0.6				96.9
A t H t A	20	0.5				97.4
H , A	1,843	48.9	A , H	532	14.1	63.0
A H	482	12.9	H A	205	5.4	81.3
H (A	146	3.9	A (H	16	0.4	85.6
A : H	94	2.5	H : A	23	0.6	88.7
H - A	66	1.8	A - H	35	0.9	91.4
A - H	31	0.8	H - A	21	0.6	92.8
H , '' A	24	0.6				93.4
H , A , A	19	0.5				93.9
A , H , A	16	0.4				94.3

Table 6.4: Apposition forms in TRAIN with abstract (top) and actual (bottom) tokens, e.g. H t A indicates an HEAD, one token then an ATTR.

Table 6.4 shows frequent apposition forms in abstract and actual forms. Most apposition units are separated by one token (75%), commonly a comma (63%), but otherwise parentheses, colons and hyphens. This matches intuitions about apposition form, but there is still a significant number (18%) that are not separated by any tokens. This suggests that while comma matching may be a strong baseline for extraction, more sophisticated models are required. Of these cases, the majority (93%) are separated by zero or one tokens.

Table 6.5 shows how entities are distributed in apposition units, using patterns of entity type and O (one or more non-entity tokens). HEADs often completely match an entity boundary, most often for PER entities. They can sometimes contain a title O PERSON or refer to multiple people, for example: the intervening tokens in most cases of PERSON O PERSON is and (e.g. Tom and Nicole). The O pattern indicates that the apposition unit contained no named entities, for example her son or the office.

HEAD	Count	%	∑%	ATTR	Count	%	∑%
PERSON	1,702	45.1	45.1	O	1,352	34.9	34.9
ORG	504	13.4	58.5	O ORG	381	9.8	44.7
O	471	12.5	71.0	O ORG O	168	4.3	49.0
O PERSON	238	6.3	77.3	O NORP O	166	4.3	53.3
GPE	90	2.4	79.7	O GPE O	115	3.0	56.3
WORK_OF_ART	83	2.2	81.9	DATE	87	2.2	58.5
MONEY	55	1.5	83.3	O GPE	87	2.2	60.8
PERSON O PERSON	32	0.8	84.2	ORG O	77	2.0	62.8
DATE	30	0.8	85.0	GPE O	75	1.9	64.7
FAC	30	0.8	85.8	O ORG O GPE	68	1.8	66.4
PRODUCT	29	0.8	86.5	CARDINAL O	51	1.3	67.8
CARDINAL O	25	0.7	87.2	O CARDINAL O	51	1.3	69.1
LOC	19	0.5	87.7	O PERSON O	48	1.2	70.3
O ORG	18	0.5	88.2	MONEY	47	1.2	71.5
O CARDINAL O	16	0.4	88.6	ORG	44	1.1	72.7
O NORP O	16	0.4	89.0	O ORG O ORG	42	1.1	73.7
GPE O GPE	15	0.4	89.4	PERSON O	41	1.1	74.8
O PERSON O	14	0.4	89.8	O GPE O ORG	38	1.0	75.8
ORG O ORG	12	0.3	90.1	NORP	37	1.0	76.7
ORG O PERSON	12	0.3	90.4	O GPE O GPE O	30	0.8	77.5

Table 6.5: The top-20 most frequent HEAD/ATTR NE tag patterns found in TRAIN. Where one or more tokens is outside an entity span, it or they are represented by O. For example, PERSON O PERSON refers to an HEAD consisting of a pair of PER entities separated by at least one non-NE token.

ATTRs most commonly feature only non-entity tokens (34.5%) and when an entity is present, it is accompanied by other text. For example, O ORG might map to chairman of ORG. HEADs are typically shorter (median 5 tokens, 95% < 7) than ATTRs (median 7 tokens, 95% < 15). This, coupled with the often lack of NE structure in ATTRs, suggests a strategy that concentrates on identifying the HEAD, then searching for

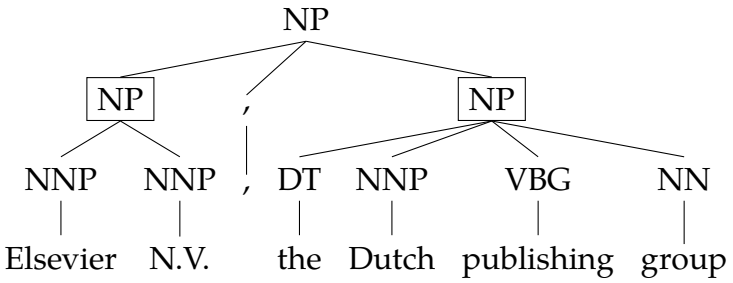


Figure 6.1: Appositional NP candidates (nw/wsj/00/wsj_0001)

suitable ATTRs. From an NEL point of view, this is the extraction of disambiguating information (ATTRs) for a previously identified entity (HEAD).

6.3 Techniques for extracting apposition

This section motivates the contributions of this chapter: three complementary techniques for more faithfully modelling apposition in systems that use a full syntactic parse. These systems use the parse to first generate candidates (pairs of NPs: p_1 and p_2), then classify them as apposition or not. System that use these techniques are presented in Section 6.4.

6.3.1 Syntactic restrictions for phrase candidate selection

An ideal candidate generation process would select only phrases that are correct apposition HEAD and ATTR units. Unfortunately, there are many cases of adjacent constituents that are not valid appositions, so a process should try and filter these from the candidates.

Parse node labels are critical—we restrict candidate phrases to those labelled NP or NML. Punctuation can separate adjacent NPs and, given a candidate phrase, we search within the sentence for phrases before and after, allowing intervening punctuation. If there is an NP with an adjacent NP to the right (not counting punctuation), both phrases are considered candidates: either HEAD or ATTR.

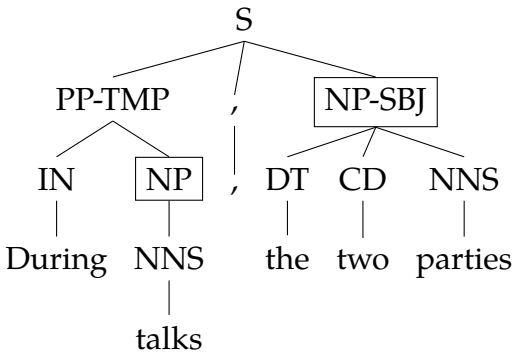


Figure 6.2: Non-appositional NP candidates (nw/xinhua/03/chtb_0300)

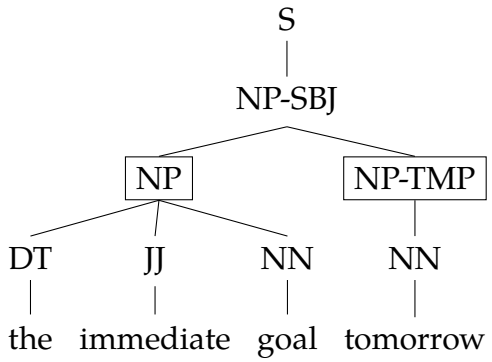


Figure 6.3: Non-appositional but restricted candidates (bn/abc/00/abc_0011).

Figure 6.1 shows a tree fragment containing an apposition. The candidate phrases are boxed. Each of them is considered a candidate since it has an adjacent NP. In this case, the first phrase is the HEAD and the second an ATTR. The tree in Figure 6.2 also shows adjacent NPs that are not appositional as they do not share a parent NP.

We propose stricter syntactic restrictions on candidate generation that better model the rules in Section 6.1.1. In the restricted setting, we only consider adjacent NPs that are sibling children of an NP, as is the case in Figure 6.1. This better matches the condition where either unit can be omitted resulting in a grammatical sentence. Unfortunately, the restrictions permit some non-apposition cases (e.g. Figure 6.3).

6.3.2 Semantic compatibility features

In apposition, an ATTR usually adds information about its HEAD. We design features to model the *type* of information contained in an ATTR. Example 2 (John Ake) has

HEAD NE type	Seed synsets	Example terms
PER	person.n.01	Australian, yachtswoman, leader
ORG	organization.n.*	securities firm, publisher
	company.n.*	cohort
	building.n.01	research center, hotel
	body.n.02	faculty, opposition
	facility.n.01	television channel, menagerie
	producer.n.03	producer
	establishment.n.04	bookstore, florist
LOC	location.n.01	township, heartland
	way.n.06	entrance
	facility.n.01	dump, airstrip
	geological_formation.n.01	valley, basin
	state.n.01	province
	state.n.04	land
	political_system.n.01	republic
	establishment.n.04	mall

Table 6.6: Seed WordNet synsets used to create the semantic gazetteers.

two ATTRS: an age and a professional description, and the HEAD is a PER entity. To capture the notion of compatible ATTRS, we extract gazetteers of valid descriptions for each entity type. WordNet (Miller, 1995) is a semantic resource where synsets, corresponding to a word's senses, are related in a graph structure. We chose a set of general seed synsets and recursively traversed the WordNet graph choosing hyponyms to build a list of specific terms. We inspected frequent ATTRS in TRAIN and chose seed synsets that yielded semantically consistent hyponyms. Table 6.6 shows the seed synsets used to build the gazetteers. We added plural forms of all terms to the gazetteers using pattern.en (De Smedt and Daelemans, 2012) to increase coverage. The simplistic traversal of the hierarchy means that the gazetteers contain noisy terms (x-axes is unlikely to occur as an ATTR for a location) and some seed synsets are more productive than others, but are a reasonable collection of descriptive vocabulary.

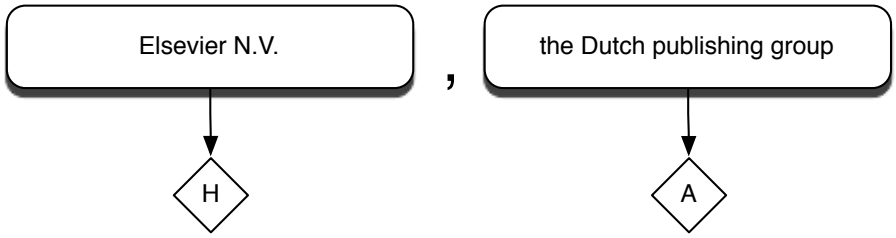


Figure 6.4: Example single phrase classification, with HEAD and ATTR the two labels.

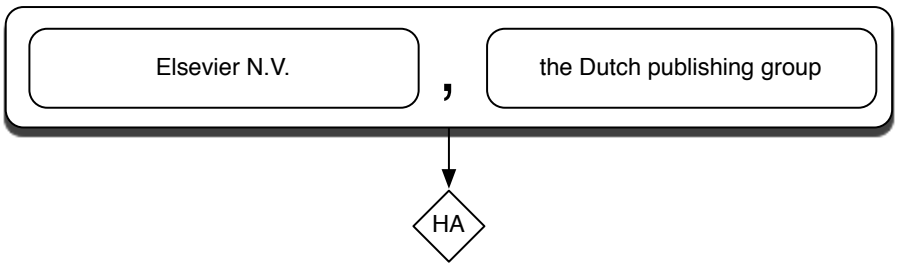


Figure 6.5: Joint classification, with HEAD-ATTR the label.

6.3.3 Joint classification

The third contribution is to classify pairs of phrases rather than a phrase in isolation. Favre and Hakkani-Tür learn an AdaBoost model[8] that combines decision stumps to classify phrases as HEAD, ATTR or neither. This does include features modelling a phrase's context: token, POS from two tokens before and after the phrase and the labels of previous and next siblings and its parent. After each candidate phrase has been classified, a filtering process removes singleton HEADs and ATTRs, leaving only well-formed apposition. Figure 6.4 shows the two classification steps applied to the tree in Figure 6.1.

A different approach is to classify pairs of phrases as HEAD-ATTR, ATTR-HEAD or NONE. Figure 6.5 shows the joint classification of the same tree. Joint modelling better captures the compatibility between the HEAD and ATTR and should lead to better results without any filtering post-process.

[8] https://github.com/benob/icsiboost

6.4 Apposition extraction systems

This section introduces apposition extraction systems used to explore the impact of syntactic restrictions, semantic features and joint prediction.

6.4.1 Pattern

Our first system uses hand-tuned lexical, POS and NE patterns to extract apposition without a full syntactic analysis. While not as accurate as a parser-based system, the pattern system scales to much larger datasets.

Extraction is a sentence-level process, each document having first been tokenised, split into sentences and tagged for parts of speech and NEs. All patterns are applied to the sentence and any of these can yield (HEAD,ATTR) tuples. While there is no interaction between patterns, the same tuple can be produced by two different patterns, but duplicates are removed before evaluation.

To develop the patterns, we examined the TRAIN dataset and attempted to generalise the statistics in Section 6.2.1 into a set of patterns. There is some overlap between TRAIN and DEV_F (44/94) and TEST_F (42/95), and some apposition from the latter two datasets would have been used in pattern generation. This is because we use the dataset split from the CoNLL 2011 task rather than the one used by FHT.

First, we defined a few basic patterns to capture common structures:

- **NP**: chains of NPs, allowing for optional determiners, adjectival modifiers and intervening conjunctions (e.g. the red car and the blue car).

- **PER, ORG, GPE**: clusters of NEs, taking account of pseudo-titles. As introduced in Section 6.1.1, these are not considered appositions by our interpretation.

- **#**: a pause represented by punctuation or interjections. Pattern-final pauses also allow: that, says and said.

- **ROLE**: tokens that occur in a gazetteer of recursive hyponyms from the WordNet (Miller, 1995) synset `person.n.01`.

- **RELATION**: tokens matching a manually-constructed gazetteer of relations: father, opponent, etc.

- **AFFILIATION** United States political party affiliation initials: D and R, representing Democrat and Republican.

Table 6.7 shows examples of the top five of the sixteen patterns ordered by recall over the TRAIN dataset. This gives some insight into the precision/recall tradeoffs that the highest coverage patterns make. Simply applying the first pattern to extract apposition allows fairly high precision (73.1%) identification of a person's professional role, leading to the highest individual F score as these have the highest recall. Unfortunately individual recall is low and we see diminishing returns as other patterns have little effect due to low coverage and despite their high precision, as is the case for the last pattern, shown in Table 6.7. Table 6.8 shows the full list of patterns. The lower end of the list consists of more specific patterns, matching rarer structures.

Following the statistics extracted from the TRAIN dataset, we concentrated on apposition that were explicitly flagged using punctuation and had ATTRs adding information to PER, ORG or GPE HEADS. One disadvantage of this development methodology is that optimising precision and recall manually can be difficult and time consuming, leading to complex pattern sets.[9]

We apply the following post-processing to filter extracted tuples. If both units are made up of the same number of NEs, with leading determiners or honorific titles (e.g. Mrs., Professor), we exclude the tuples. This is designed to stop mis-recognition of NE lists, also delimited by commas, but is relaxed for some patterns where this is valid (e.g. PER and ORG in $\{\text{Sen. Sam Nunn}\}_h$ $(\{\text{D.}\}_a)$. Unigram ATTRs are checked to ensure that they are honorific titles, numbers, relations or NORP entities. Finally, to account

[9] Available at http://schwa.org/projects/resources/wiki/Apposition

Pattern and Example	P	R	F
{PER}$_h$ # {NP (IN LOC|ORG|GPE)?}$_a$ # {Jian Zhang}$_h$, {the head of Chinese delegation}$_a$,	73.1	21.9	33.7
{DT JJ? (ROLE|RELATION)+}$_a$ #? {PER}$_h$ {his new wife}$_a$ {Camilla}$_h$	45.9	9.5	15.8
{ORG|GPE}$_h$ # {DT NP}$_a$ # {Capetronic Inc.}$_h$, {a Taiwan electronics maker}$_a$,	60.4	6.0	10.9
{NP}$_a$ # {PER}$_h$ # {The vicar}$_a$, {W.D. Jones}$_h$,	33.7	4.5	7.9
{PER}$_h$ # {NP POS NP}$_a$ # {Laurence Tribe}$_h$, {Gore 's attorney}$_a$,	82.0	4.0	7.7

Table 6.7: The top-five patterns by recall in the TRAIN dataset. # is a pause (e.g. punctuation), | a disjunction and ? is an optional part. Patterns are used to combine tokens into noun phrases for the NP symbol.

for annotation inconsistencies, the token old is removed from the end of the ATTR span if the first token is a number—{20 years old}$_a$ would be normalised to {20 years}$_a$ old.

6.4.2 Adjacent NPs

This baseline system assumes that all candidate pairs (using the restricted or unrestricted candidate generation) are apposition and is thus has low precision, but high recall as it allows all candidates to be generated. This results in conflicting apposition with a pair of phrases being labelled HEAD-ATTR and vice versa.

6.4.3 Rule

We formalise semantic compatibility as a set of rules that, for any candidate pair, require the HEAD to have a syntactic head that is part of a PER, ORG, LOC or GPE NE. The syntactic head of the ATTR must match one or more of the gazetteers from Section 6.3.2 dependent on the NE type of the HEAD.

Pattern and Example	
{PER}$_h$ # {NP (IN LOC|ORG|GPE)?}$_a$	{Jian Zhang}$_h$, {the head of Chinese delegation}$_a$,
{DT JJ? (ROLE|RELATION)+}$_a$ #? {PER}$_h$	{his new wife}$_a$ {Camilla}$_h$
{ORG|GPE}$_h$ # {DT NP}$_a$ #	{Capetronic Inc.}$_h$, {a Taiwan electronics maker}$_a$,
{NP}$_a$ # {PER}$_h$ #	{The vicar}$_a$, {W.D. Jones}$_h$,
{PER}$_h$ # {NP POS NP}$_a$ #	{Laurence Tribe}$_h$, {Gore 's attorney}$_a$,
{NE}$_h$ ({NE IN NE}$_a$)	{PASOK}$_h$ ({Mr. Papandreou's party}$_a$)
{PER}$_h$ # {ORG NP}$_a$ #	{Mr. Trudeau}$_h$, {a Writers Guild member}$_a$,
{RELATION}$_a$ # {PER}$_h$ #	{His mother}$_a$, {An Qi}$_h$,
{PER}$_h$ # {DATE|NORP}$_a$ #	{Judge Ramirez}$_h$, {44}$_a$,
{PER}$_h$ # {CD}$_a$ # {NP}$_a$ #	{Cornel Wilde}$_h$, 74 , {actor and director}$_a$,
{DT NORP? NP}$_a$ {ORG|GPE}$_h$	{the capital}$_a$ {Harare}$_h$
{PER}$_h$ # {NP}$_a$ WHO	{Barnabas de Bueky}$_h$, {a 55 - year - old former Hungarian refugee}$_a$ who
{PER}$_h$ # {NP VBN TO VB NP}$_a$ #	{Cassim Saloojee}$_h$, {a veteran anti-apartheid activist on hand to welcome Mr. Sisulu}$_a$.
{PER}$_h$ # {NP OF VBG NP}$_a$ #	{Motion Picture Association President Jack Valenti}$_h$, {the most vociferous opponent of rescinding the rules}$_a$.
{PER}$_h$ ({AFFILIATION}$_a$	{Sen. Sam Nunn}$_h$ ({D.}$_a$, Ga.)

Table 6.8: Full set of apposition extraction rules.

We use head-finding rules from Collins (1999) adapted to handle NML phrases (Vadas and Curran, 2007), and partitive constructions by checking if the first token of the NP is one, some, many or a number (e.g. inside the PP, we recover students from one of the students). PER HEADS must have ATTRS whose syntactic head matches the PER gazetteer. To handle idiosyncrasies of age constructions, we allow ATTRS whose initial token is composed of fewer than 3 digits and contains years if the ATTR is longer than one token. ORG and LOC/GPE HEADS must have syntactic heads that match the

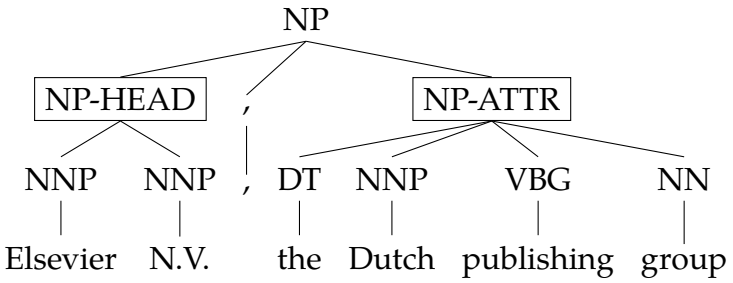


Figure 6.6: Labelled appositional NP candidates (nw/wsj/00/wsj_0001)

ORG and LOC gazetteers respectively or the PER gazetteer to account for metaphoric description (e.g. the champion, Apple). In addition, the head of their ATTRs must not contain ORG, FAC, LOC, GPE or DATE NEs, except in the case of ORG HEADs where a left parenthesis or bracket separates it from the ATTR.

The resulting pairs of apposition units are re-ordered by specificity, as per Section 3.1 of the OntoNotes coreference guidelines. The scale ranks from high to low: proper nouns, pronouns, definite NPs, indefinite NPs, and finally NPs. This is judged by presence of appropriate POS tags and a list of definite and indefinite articles. If the HEAD is of lower specificity than the ATTR, the labels are reversed. The rare case of multiple ATTRs is handled as follows: if an ATTR has been extracted, the rules are applied to the next candidate ATTR in the same direction.

6.4.4 Labelled Berkeley Parser

This system uses a Berkeley Parser (Petrov and Klein, 2007) model trained on trees whose noun phrases have been labelled with HEAD and ATTR to match gold-standard apposition as in Figure 6.6. During evaluation, sentences are parsed and candidate NPs labelled with HEAD and ATTR are marked as appositions. When no syntactic constraints are applied (LBP_F), this is equivalent to FHT's LBP system (LBP_F, marked †).

6.4.5 Phrase classification

Each NP is independently classified as HEAD, ATTR or NONE. We use a log-linear model with a stochastic gradient descent optimizer from SCIKIT-LEARN (Pedregosa et al., 2011), using default options and a standard feature scaler without a centred mean. The binary features are calculated from a generated candidate phrase (p) and are the same as FHT's phrase system:

- 1- to 4-grams of phrase tokens and POS tags
- Labels of the phrase, its parent and siblings
- 1- to 4-grams of tokens and POS tags before and after the phrase
- NE labels within the phrase and the label of a NE that matches the phrase bounds
- Punctuation/interjection before or after
- If the first token has the POS tag DT or PRP$
- Semantic gazetteers matching p_1's head
- If p_1's head has the POS CD
- p_1's NE type
- Specificity rank

If two adjacent NPs are classified with one label as HEAD and the other as ATTR, they are proposed as an apposition and the re-ordering process applies. Replications of FHT's phrase system are marked in tables with ‡ (Phrase_F) and ⋆ (Phrase_F using labelled trees during prediction).

6.4.6 Joint classification

The final system learns a model, again using a log-linear model, to classify *pairs* of phrases into three classes: HEAD:ATTR, ATTR:HEAD and NONE (filtering is not necessary).

The features from the phrase system are calculated for each phrase and marked with which phrase they were extracted from. The additional features for the joint model are calculated as follows:

- Compound features from the cross-product of gazetteer, specificity, CD and NE phrase syntactic head features. For example, a pair may yield the feature `gaz-per-headne-per`, encoding a semantic compatibility between a PER NE in the syntactic head of one phrase and the membership of the other phrase's syntactic head in the PER gazetteer.
- Compound features indicating any of the above features found in both phrases.
- Whether the phrases have equal specificity or which was more specific.
- Whether the text of one phrase can be considered an acronym of the other (e.g. $\{\mathsf{UN}\}_{\mathsf{a}} \rightarrow \{\mathsf{United\ Nations}\}_{\mathsf{h}}$).

6.5 Results

We evaluate by comparing the extracted HEAD-ATTR pairs against the gold standard. Pairs must match both the gold-standard bounds and label to be correct. We report precision (P), recall (R) and F-score (F). Taggers and parsers are trained on TRAIN and evaluated on DEV or TEST. We use the C&C POS and NE taggers (Curran and Clark, 2003) and the Berkeley Parser (Petrov and Klein, 2007), all with default parameters.

Table 6.9 shows initial results over the DEV dataset. The top half of the table details reimplementations of three FHT systems: LBP_F, Phrase_F and Phrase_F LBP. The results show a similar trend to Favre and Hakkani-Tür (2009) (see Section 6.1.3) as their best system combines the Phrase_F model with the LBP trees, scoring 53% F, largely due to the high precision of 78%. We do not train Phrase_F with labelled trees as this would require n-fold parser training to provide the Phrase_F model with automatic trees. We believe that the reimplementations provide a fair comparison, showing the impact of combining the labelled tree with a statistical model.

System	P	R	F
LBP$_F$	48	53	†50
Phrase$_F$	77	40	‡52
Phrase$_F$ LBP	**78**	40	⋆53
Pattern	45	35	39
LBP	66	52	**58**
Adj NPs	12	**58**	19
Rule	65	47	55
Phrase	73	45	56
Joint	67	49	56

Table 6.9: Results over DEV: FHT replications (top) and our systems (bottom). Highest values in each metric are in **bold**.

The bottom section are our system results using syntactic restrictions, semantic features and, where appropriate, standard Berkeley Parser trees. Pattern performance is reasonable at 39% F given its lack of full syntactic information. Our reimplementation of FHT's LBP with syntactic restrictions extracts appositions from the labelled Berkeley Parser trees and performs the best at 58% F. The performance increase over LBP$_F$ can be explained by the NP parent and sibling restrictions. This accounts for less than 1% in recall, but has the benefit of an almost 20% increase in precision. The Adjacent NPs baseline has the highest recall (58%), but low precision as it assumes that all phrase pairs are in apposition. As such, it is an upper bound to parse-based systems that use unlabelled Berkeley Parser trees. Statistical models improve performance over simple rule-based application of the semantic features (55%), with the joint model (56%) similar to the single phrase model (56%). Although the single phrase model has context features, it must make two independently correct classifications, where the joint model classifies the pair—despite the former's higher precision.

Table 6.10 presents an ablative analysis showing the relative effect of two of our contributions: syntactic restrictions and semantic features. The left column (grey) is as in Table 6.9 and the columns to the right show system performance without semantic

System	Full			-sem			-syn			-sem -syn		
	P	R	F	P	R	F	P	R	F	P	R	F
Pattern	45	35	39	-	-	-	-	-	-	-	-	-
LBP	66	52	58	-	-	-	48	53	†50	-	-	-
Adj NPs	12	58	19	-	-	-	4	65	7	-	-	-
Rule	65	47	55	-	-	-	44	50	47	-	-	-
Phrase	73	45	56	72	46	56	78	41	54	77	40	‡52
Joint	67	49	56	65	49	56	71	49	57	64	48	55

Table 6.10: Results over DEV, ablative analysis. Left columns from Table 6.9 in grey and the remaining show results: without semantic features, without syntactic restrictions and with neither.

System	Full			+LBP			+gold		
	P	R	F	P	R	F	P	R	F
Pattern	45	35	39	-	-	-	52	40	45
Adj NPs	12	58	19	12	**59**	20	16	**85**	27
Rule	65	47	55	65	47	54	79	62	70
Phrase	73	45	56	**75**	46	57	**86**	58	70
Joint	67	49	56	70	51	**59**	87	68	**76**

Table 6.11: Results over DEV, with LBP trees and gold resources.

features, syntactic restrictions and both. Two replicated FHT systems are marked: LBP_F (†) and Phrase_F (‡). Our contributions result in an increase of approximately 8% and 4% F over the LBP_F and Phrase_F models respectively.

Syntactic restrictions have a greater effect than semantic features. There is some complex interaction between them, as removing both seems to perform better than simply removing the syntactic restrictions. Removing semantic restrictions on the joint model also improves performance by 1% to 57% F. This is due to a higher precision and perhaps some phrase pairs excluded by the restrictions are important for training the joint model.

As well as varying candidate selection and features, we can also explore the use of different resources passed to the models during extraction. Table 6.11 again includes the full results, adding columns for performance using the labelled trees (LBP) and gold-standard resources.

Labelled trees are produced by the apposition-aware parser model and result in performance increases for all our parse-based systems except Rule. This is consistent with Favre and Hakkani-Tür's results, indeed our reimplementation also benefitted from labelled trees, scoring 53% F (Phrase_{F} LBP in Table 6.9). Our best system combines the joint model with labelled trees for an F of 59%–6% above the best FHT performance.

While using gold-standard trees, NE and POS tags is not a realistic setting, it reveals how robust each system is to error derived from using their automatic equivalents. The Pattern system is fairly robust to this error, dropping only 6% F when using automatic tags. Adjacent NPs drops slightly more, but still has imperfect recall. This places an upper-bound on recall for systems using syntactic restrictions: 85% assuming perfect parse trees, 59% using labelled trees. There is still a significant number of gold-standard trees that are not captured by the syntactic restrictions as we might expect 100% recall using gold-standard parse trees.

Possible explanations are that some missing cases are adverbial phrases, which we do not handle, or errors in the gold-standard annotation that contradict the syntax. The remaining systems fare worse when using automatic trees and tags, between 13% and 17%, suggesting parser error has a significant impact on apposition extraction performance. As the joint LBP model uses parse trees and NE tags, we experiment with two extra settings: LBP trees and gold NE tags, and gold trees and auto tags. The former drops 16% F relative to the both-gold setting and the latter only 1% F. This confirms that our model relies heavily on the syntactic information provided by correct parses to extract apposition.

Another contribution of this work is to provide more empirical analysis of how well apposition can be extracted in different contexts. Favre and Hakkani-Tür's emphasis on ASR output necessarily restricted their work to speech, where we evaluate on

Subcorpus	Model	P	R	F
Broadcast conversation	Pattern	49	21	29
	Joint LBP	65	31	42
Broadcast news	Pattern	40	37	38
	Joint LBP	60	44	50
Magazine	Pattern	38	40	39
	Joint LBP	63	53	58
Newswire	Pattern	49	40	44
	Joint LBP	78	65	71
Web	Pattern	40	23	29
	Joint LBP	43	21	28

Table 6.12: Results for Pattern and Joint LBP on the different domains in DEV.

text and speech. Table 6.12 shows how two representative systems: Pattern and Joint LBP, perform across the dataset. Joint LBP performs better than the Pattern system in broadcast, magazine and newswire. Apposition extraction was the most successful at 71% F on newswire—where the benefit of parse trees was most evident with a 27% improvement in F. One possible explanation is that we train the Berkeley Parser on all data and, as newswire is the largest at 38%, its statistical model is more adapted to the newswire domain. The alternative would be to train a parser model for each source, but only some sources have sufficient amounts of data for training. The worst results are in web, where parse trees *diminish* performance by 1% points F. This may be because newswire is also bound by a more formal style than open web text and apposition, as we discuss in Section 6.1.1, serves specific purposes in reportage, occurring more frequently. In conclusion, it is worthwhile using a parse-based model for extracting apposition from newswire. In the web domain, a POS and NE Pattern system not only extracts apposition more accurately, but also avoids the computational costs of syntactic parsing.

System	Full			+LBP		
	P	R	F	P	R	F
LBP$_F$	-	-	-	52	38	†44
Phrase$_F$	64	29	‡40	68	34	⋆45
Pattern	57	47	52	-	-	-
LBP	-	-	-	62	38	47
Adj NPs	10	47	16	11	**54**	19
Rule	61	28	38	70	38	50
Phrase	59	29	39	**72**	38	50
Joint	58	35	44	65	46	**53**

Table 6.13: Results over TEST$_F$ with regular and labelled parse trees.

System	Full			+LBP		
	P	R	F	P	R	F
LBP$_F$	-	-	-	53	47	†50
Phrase$_F$	71	32	‡44	71	33	⋆45
Pattern	42	31	36	-	-	-
LBP	-	-	-	64	45	53
Adj NPs	11	**57**	18	10	**57**	17
Rule	63	41	50	61	41	49
Phrase	72	38	50	**73**	39	51
Joint	65	43	52	67	45	**54**

Table 6.14: Results over TEST with regular and labelled parse trees.

We also evaluate on the dataset replicated from Favre and Hakkani-Tür (2009), training on TRAIN$_F$ and testing on TEST$_F$. Table 6.13 shows the full system results and using labelled trees. Our reimplementations of the systems from Favre and Hakkani-Tür (2009) perform better than their reported results and Phrase$_F$ LBP scores 45%. Again, the Joint LBP system is the best of our systems at 53%, 8% F above the FHT systems. The Pattern system is surprisingly competitive at 52% F, substantially higher than performance on DEV.

Finally, we evaluate the best of our systems on the OntoNotes 4 TEST dataset and these follow the same trend as DEV. Table 6.14 shows that joint LBP performs the best at 54%, 4% above LBP_F. We test whether this difference is statistically significant using a bootstrap test (Efron, 1979). We run 10,000 trials where documents are sampled with replacement from the test corpus. Each trial, we evaluate both systems and count those trials where the F difference is greater than or equal to the difference originally observed. This occurs once and our p-value is 0.00, showing that the difference between our and FHT's system is statistically significant ($p < 0.01$).

Labelled trees help apposition extraction and perhaps due to some more detailed syntactic analysis of how noun phrases relate to one another. We test whether labelling apposition on trees can help parsing. We parse DEV trees with LBP and BP, remove apposition labels and analyse the impact of labelling using the Berkeley Parser Analyser (Kummerfeld et al., 2012). Table 6.15 shows the LBP makes fewer errors, particularly NP internal structuring, PP and clause attachment classes at the cost of modifier attachment and co-ordination errors. Rather than increasing parsing difficulty, apposition labels seem complementary, improving performance slightly. It should be noted that this is only a 1% error reduction and unlikely to be statistically significant, nonetheless, explicitly modelling apposition is a linguistically-principled way to improve parser performance.

Error	BP	LBP	δ
Modifier Attachment	1,523	1,700	177
Co-ordination	3,095	3,245	150
NP Attachment	2,615	2,680	65
PP Attachment	5,585	5,396	-189
NP Internal Structure	1,483	1,338	-145
Clause Attachment	3,960	3,867	-93
Different label	2,960	2,904	-56
VP Attachment	1,148	1,095	-53
Single Word Phrase	2,872	2,819	-53
Unary	1,784	1,751	-33
Other	3,164	3,064	-100
TOTAL	30,189	29,859	-330

Table 6.15: Selected BP/LBP parse error distribution.

6.6 Summary

Writers use apposition to precisely communicate information about entities, which could be used to disambiguate them. This chapter presents three apposition extraction techniques and systems that use them. Linguistic tests for apposition motivate strict syntactic constraints on candidates and semantic features encode the addition of compatible information. Joint models more faithfully capture apposition structure and our best system achieves state-of-the-art performance of 54%. Parsing accuracy is critical for apposition extraction, but chunking may provide robust and efficient candidate generation. As Favre and Hakkani-Tür note, chunk boundaries are not a good match for apposition, but labelling chunks with apposition may help. Our results will immediately benefit the large number of systems with apposition extraction components for coreference resolution and IE. In the next chapter, we explore how apposition and other locally described attributes can improve entity linking.

7 Local description for NEL

> Just weeks after the Australian Crime Commission released a damning report warning of the threat of match fixing in sport, the case involving {rugby league identity}$_d$ {John Elias}$_e$, {former Parramatta player}$_d$ {Brad Murray}$_e$, and {Jai Ayoub}$_h$, {the son of Murray's manager Sam Ayoub}$_a$, will be before the Downing Centre Criminal Court on Wednesday.

Apposition, local description and lists in McClymont, SMH 2013-02-15[1].

Our state-of-the-art NEL system uses the KB structure and a mention's context. These techniques perform well at linking mentions to KB entries and we use simple techniques such as thresholding and rules to identify and cluster NIL mentions. While a mention's document or sentence context may contain precise descriptions of entities, it will typically contain other content, adding noise to any similarity calculation. Our goal is to take advantage of the same contextual cues that readers do to resolve name ambiguity. When authors describe entities in the discourse, the content and manner in which they do this can help disambiguate their mentions.

We explored apposition in Chapter 6 and found it surprisingly difficult to extract. We begin this chapter by using apposition for linking, finding that it does not have much impact when added to our unsupervised system from Chapter 5. While apposition is a key way in which authors describe entities, it is not the only one.

[1] www.smh.com.au/lifestyle/fill-im-up-lunch-is-on-them

This chapter takes a more general approach to capturing a mention's *local description*, which we define as the specification of an entity attribute, typically a noun phrase, in the same sentence as its mention. We hypothesise that these descriptions can help provide positive and negative evidence for linking.

(1) Former Prime Minister John Howard spoke in favour of the legislation.
(2) Local plumber John Howard slammed the new legislation.

We can extract roles from the sentences in Examples 1 and 2: Former Prime Minister and Local plumber. In assessing a match between each mention and the candidate `John Howard`, the role Former Prime Minister should match the article text, where Local plumber will probably not. The remaining sentence context, legislation will also be a fairly strong match for the political candidate, so local description is important. Systems already take advantage of a match in the former, but should be able to take advantage of the *absence* of any match in the latter.

Our first contribution is an analysis of the TAC 11 queries that characterises where local description that might help disambiguation is found. Our semantic and syntactic analysis examines the type of attributes in descriptions and how they are specified. We find that KB and NIL mentions differ in how they are described. Our second contribution is a technique for modelling these as features in our state-of-the-art linking system, accounting for their use as positive and negative evidence. We conclude with an overview of some of the key problems and directions for future research.

7.1 Apposition for NEL

We motivated apposition extraction with the task of identifying entity attributes for disambiguation. In practice, this amounts to matching a NE mention's attributes in a context document with the correct candidate's KB entry. This is often modelled as cosine similarity between the two texts, but that may also result in incorrect candidates

System	Dataset	Accuracy			B^{3+} F		
		All	KB	NIL	All	KB	NIL
Document	11	87.6	82.1	93.1	84.6	80.4	88.8
Document +sentence	11	87.4	81.3	93.5	84.4	79.6	89.1
Document +sentence +apposition	11	86.8	79.3	94.3	83.5	77.3	89.8

Table 7.1: Impact of increasingly local similarity scope for NEL on TAC 11.

scoring highly through spurious matches. Our goal is to extract a mention's precise, local context, appositive attributes and use them for disambiguation.

We use the unsupervised linker from Chapter 5 to investigate whether using information close to the mention is more useful for disambiguation. This system simply averages feature values and so higher similarities will boost a candidate's score.[2] Table 7.1 shows the KB, NIL and overall accuracies over TAC 11 queries using similarity features at a document, sentence and apposition scope. The document feature calculates the cosine similarity between unigram counts from the context document and the Wikipedia text of the candidate entity. The sentence feature restricts the bags of words to unigrams from sentences that contain the mention's coreferent names. Apposition is a boolean feature—whether any tokens in a mention's apposition attribute, extracted using the Pattern system, exist in the Wikipedia text. Adding sentence-level similarity decreases performance from 87.6% accuracy to 87.4%, while adding apposition matching further reduces it to 86.8%. Since we use the Pattern extraction system, noisy extraction is certainly an issue, but in this is outweighed by the performance benefits as a full syntactic parse is not required. Even still, apposition ATTRs were only extracted for any mention in the query chain for 134 of the 2,250 queries. As a result, coverage is too low and extraction too noisy to have a positive impact, especially from such a high baseline.

[2] Although each individual feature's effect is "diluted" as more features are used

7.2 Describing entities

We aim to extract entity descriptions and use them for disambiguation. There is a substantial body of work concerned with describing entities. Relation extraction is typically concerned with relations between two entities. In Chapter 2, we introduced an example from Freebase of a structured relation in Table 2.1:

(3) `<John Howard, /govt_positions_held, Prime Minister of Australia>`

This fact indicates that a specific John Howard held a specific post: the relation between the two entities. Relation extraction requires that they be extracted from text, with different levels of structure. Systems typically use varying levels of supervision from fully supervised, to seeded bootstrapping and unsupervised learning. Distant supervision (Mintz et al., 2009) uses Freebase relations to extract training data from Wikipedia, which is used to learn relation classifiers (e.g. a classifier for `/govt_positions_held`). Their model uses lexical, syntactic and NE mention features, and they find syntactic features especially useful. The TAC slot filling task (Ji et al., 2011) formalises a relation extraction task for KBP. It specifies a fixed schema of attributes, or slots, that must be extracted for a query entity. Once each likely attribute is extracted, their slot must be identified. Producing the final value may require normalisation or inference.

As discussed in Chapter 2, entity attributes have been used directly for resolving name ambiguity. Mann and Yarowsky (2003) use bootstrapped rules to extract biographic facts from text for clustering mentions. The disambiguating terms learned in the generative model of Li et al. (2013) are descriptions that also have discriminative power over a set of ambiguous candidates. The closest work to our goal is Cheng and Roth (2013), whose joint model accounts for relations and links. In particular, they take advantage of apposition and pseudo-titles (e.g. local plumber) to extract a precise mention description. They take the cosine similarity between an extracted ATTR and

the candidate article as a signal whether a candidate "entails" the mention form. This is similar to the work in apposition for textual entailment reviewed in Chapter 6.

While document-wide context similarity has been popular in TAC systems, alternatives have been explored: slot value context (Cassidy et al., 2011), infobox fields (Clarke et al., 2012) and limited token windows (Fahrni et al., 2011, 2012). Systems have explicitly targeted other mentions, extracting relations (Bonnefoy and Bellot, 2012; Clarke et al., 2012) and searching for a candidate's "appositional"[3] terms in the text such as `Illinois` for `Toronto, Illinois` (Graus et al., 2012; Clarke et al., 2012).

Our baseline approaches to modelling context do not rely on exact matches, using cosine similarity between bags of words from the document (`unigram_cosine` and `bigram_cosine`) and mention sentences (`sent_cont`). Rather than attempt to capture precise attributes in text and match them exactly, we opt instead to *restrict* a mention's context for matching. Restricted context should minimize spurious similarity between document and candidate articles, improving KB and NIL matching. Long and detailed candidate articles may mention many entities and general concepts and so within a list of candidates, some context will be shared among them. For example both `John Howard` and `John Howard (Australian Actor)` will contain some shared content such location mentions, as they were both born in `New South Wales, Australia`.

Restricting context for matching allows us to check explicitly for the absence of a match. Specifically, if we extract an entity attribute and it is *not* found in the candidate article, then this is evidence against that candidate. Modelling non-matching is only appropriate when we can restrict context to where an attribute is most likely to be describing its mention. This is because of spurious similarity, for example from an incidental mention of New South Wales, possibly far from the mention.

We use n-gram overlap between a mention's local descriptions and different fields of the candidate Wikipedia article. Thus, in contrast to other approaches, we do not require exact identification of the attribute spans or classification into specific types as in slot filling. While this could be advantageous, it is not the focus of this work.

[3]Note that this is *not* apposition as defined in Chapter 6.

We also capture descriptions for ORG and LOC mentions, where the task in Mann and Yarowsky (2003) only requires personal biographic details. Finally, our local description matching features explicitly model the presence of local descriptions and the lack of a candidate match.

7.3 An analysis of local description

The goal of this analysis is to explore mention descriptions for disambiguation: what attributes, how they are described and how it helps disambiguation. Mentions are described in different ways depending on their type and descriptions vary in semantic type and syntactic realisation. We focus on the broad sets of syntactic and semantic classes useful for disambiguating entities. We examined the 2,250 queries from TAC 11, checking for local description that could possibly help disambiguation. Each query's context document was processed by segmenting sentences, tagging NEs and performing naïve in-document coreference as described in Chapter 5. This allowed us to check each sentence which the query term's NE appeared in for local descriptions. A single annotator[4] examined all query sentences, answering the question: *"Does this sentence contain local descriptions that might help you to disambiguate the entity?"*. Where the annotator decided that disambiguation would be difficult even with the description, they would mark it questionable. We did not annotate the description tokens and thus extraction accuracy cannot be directly evaluated.

Table 7.2 gives an overview to the detailed description below. In examples, the query's entity is marked `e` and the description `d`. The next sections discuss each mention type in more detail.

7.3.1 Locations

A location mention can be described with different attributes: what it is, where it is and other miscellaneous details.

[4]The author.

NE type	Attribute	Example
LOC	address	{Vineyard Community Church}$_e$, {Millersville, MD}$_d$
	desc	{Gilroy}$_e$, {once the aromatic apogee of garlic}$_d$,
	orient	{Dawr}$_e$, a town {southeast of Tikrit}$_d$
	place	{Abbotsford}$_e$, {British Columbia}$_d$
	is place	{New Haven}$_d$, {Connecticut}$_e$
	type	the {city}$_d$ of {Tokushima}$_e$
ORG	address	{Features Department}$_e$, {Star-Telegram, Box 1870, ...}$_d$
	alias	{Federal Highway Police}$_e$ ({PRF}$_d$)
	place	{University Teaching Hospital}$_e$ in {Lusaka}$_d$
	sponsor	{Rupert Murdoch}$_d$'s {News Corp.}$_e$
	type	German {ARD}$_e$ {television}$_d$
PER	age	{Feldmayer}$_e$, {50}$_d$,
	place	{Fabian Hambuechen}$_e$ of {Germany}$_d$
	relation	{Driscoll's widow}$_d$, {Adelaide}$_e$
	sponsor	The {ACLU}$_d$'s {Sparapani}$_e$
	type	{shooting guard}$_d$ {Liu Xiangtao}$_e$

Table 7.2: Overview of local description types

Address A full address can be specified in noun phrases to the right. The query in this case required resolving the name TMC to Thomas Merton Center.

(4) {Thomas Merton Center}$_e$, {5125 Penn Avenue, Pittsburgh 15224}$_d$

Description This category is loosely defined and can include population demographics or refer to a notable event that occurred there. These descriptions may also contain other attributes: Example 5 includes the location type, a town.

(5) The racial tensions leading to the fight began in August in {Jena}$_e$—{a town of 2,900 with about 350 black residents}$_d$—... (an apposition)

(6) But witnesses and officials said those killed were members of a family that had sought refuge in {Dawr}$_e$, {where former President Saddam Hussein was captured in 2003}$_d$. (a relative clause)

Orientation This description offers some reference point for locating the mention that is not some kind of containing region.

(7) ...the building they attacked in {Dawr}$_e$, {a town southeast of Tikrit}$_d$. (an apposition)

(8) {Bessemer}$_e$ is {about 15 miles (24 kilometers) southwest of Birmingham}$_d$.

(9) ...in {Villepinte}$_e$, {northeast of Paris}$_d$. (an NP to the right)

(10) ...{Esmeraldas}$_e$ {in northern Ecuador}$_d$, {close to the Colombia border}$_d$, (a PP to the right)

(11) ...{Plattsburgh}$_e$, {which is about 15 miles from Canada in eastern New York}$_d$. (a relative clause to the right)

Place These describe a *containing* location, often using an NP or PP to the right, prefixed with a comma. The latter case seems similar to apposition, as it is marked by commas. We do not consider this apposition, however, as the two mentions do not refer to the same entity (e.g. Abbotsford is found in the region British Columbia in Example 12).

(12) {Abbotsford}$_e$, {British Columbia}$_d$ (an NP to the right, comma-prefix)

(13) {VICTORIA}$_e$ ({Seychelles}$_d$) (an NP to the right)

(14) ...{La Union}$_e$ city {in Zacapa province}$_d$ ... (an PP to the right)

Is a place A location can appear in the description of another mention, as its place. In the examples below, the entity of interest is in the description, but this relation may help disambiguation.

(15) ...but then again, {Floyd}$_e$, {VA}$_d$ is a different kind of place...

(16) ...professor at the {University of Pittsburgh}$_e$ at {Johnstown}$_d$ (in a PP)

(17) ...{John O'Donnell}$_e$, {a 15-0 welterweight from Croydon}$_d$, ...[5]

Type A location's type can be described as part of an apposition or noun phrases to the left or right. The examples below all describe what each of the locations is—a city.

(18) a bus from Shanghai to {Lichuan}$_e$, {a small city in the central province of Hubei}$_d$, (an apposition)

(19) {the western coast city}$_d$ of {Montecristi}$_e$. (an NP to the left)

(20) ...{La Union}$_e$ {city}$_d$ in Zacapa province... (a NP to the right)

7.3.2 Organisations

Organisation descriptions include addresses, aliases, places, sponsors and types.

Address These are found in NPs to the right.

(21) ...{Features Department}$_e$, {Star-Telegram, Box 1870, Fort Worth...}$_d$

Alias Organisation aliases can be described using acronyms or apposition.

(22) ...a trade expert with the {Ministry of Commerce}$_e$ ({MOC}$_d$) has said. (an acronym)

(23) The mainstream {Mormon Church}$_e$, {or the Church of Jesus Christ of Latter-day Saints}$_d$, ... (an apposition)

Place An organisation's place is typically a location found inside an apposition or an NP or PP to the left or right.

[5]The query entity in this example is Croydon, the boxer's hometown.

(24) . . . the {Badr Organization}$_e$, {the paramilitary of the Supreme Council for Islamic Revolution in Iraq}$_d$. (an apposition)

(25) . . . published on Friday by the {French}$_d$ weekly {Le Point}$_e$.

(26) {Singapore}$_d$'s {Ministry of Health}$_e$ (MOH) said Saturday. . .

(27) . . . statistics released by the {Immigration Office}$_e$, {Tribhuvan International Airport}$_d$ . . . (an NP to the right)

(28) The {High Judicial Council}$_e$ of {Libya}$_d$ is to convene on Monday and has the power to free the six. (a PP to the right)

Sponsor Organisations can be "sponsored" by other organisations or people. In the case of NPs to the left, the query is often only part of a structured name, for example [Army [Inspector General's]] office in Example 29.

(29) . . . an investigation by the {Army}$_d$ {Inspector General's office}$_e$.

(30) {The Tax Policy Center}$_e$, {a think tank run jointly by the Brookings Institution and the Urban Institute}$_d$, concluded . . . (an apposition)

(31) {Peruvian Supreme Court}$_d$'s {Special Penal Hall}$_e$ (SPE) on Wednesday decided. . . (a possessive to the left)

(32) A staff with the {Institute of Hydrobiology}$_e$ {of the Chinese Academy of Sciences}$_d$ . . . (an PP to the right)

Type An organisation's type can be specified using apposition, NPs to the left and NPs and clauses to the right.

(33) . . . the {Badr Organization}$_e$, {the paramilitary of the Supreme Council for Islamic Revolution in Iraq}$_d$. (an apposition)

(34) ...Swiss newspapers angrily called on former top managers of {banking giant}$_d$ {UBS}$_e$ to return bonuses ... (an NP to the left)

(35) The {NHL}$_e$ {is the greatest league in the world}$_d$ ... (a copula to the right)

(36) ...cameraman who works for the German {ARD}$_e$ {television}$_d$. (a NP to the right)

(37) ...president of the {Border Trade Alliance}$_e$, {which represents businesses all along the border with Mexico}$_d$. (a relative clause to the right)

7.3.3 People

The local descriptions for a person can include the following attributes: age, place, sponsor, type and relation.

Age Age is expressed as an apposition-like structure or an NP, or adjectival phrase to the left of the entity. As they are expressed relative to the document's date of publication, some inference would be required to transform the relative age to one of two absolute years of birth for disambiguation.

(38) {Carney}$_e$, {49}$_d$, said he didn't know whether Obama or Clinton would be better for his re-election bid. (apposition)

(39) British police say {40-year-old}$_d$ {Ali Behesti}$_e$, {22-year-old}$_d$ {Abrar Mirza}$_e$, and {30-year-old}$_d$ {Abbas Taj}$_e$ are charged with plotting to endanger life and damage property.

Example 39 is tricky—at first glance, one might assume the commas to be flagging an apposition, but in this case, they are used to list the three people, with age specified as NPs to the left of the mention.

Place A person's place—location, origin or ethnicity—can also be described using a wide range of syntactic forms: apposition, possessive and unmarked NPs to the left and NPs and PPs to the right. The description must include a LOC entity.

(40) called on Vice President {Tariq al-Hashemi}$_e$, {the lone Sunni Arab invited to the talks}$_d$. (apposition)

(41) {Frenchman}$_d$ {Alain Bernard}$_e$ held... (an NP to the left)

(42) ... and {Australia}$_d$'s {John Howard}$_e$ (a possessive to the left)

(43) 2: {Ryan Dodd}$_e$, {Canada}$_d$, 64.30. (an NP to the right)

(44) Fifty-year-old {Sidney Walker}$_e$, of {White Oak}$_d$, ... (a PP to the right)

Relation Relations to other people, either familial or otherwise (e.g. colleague), can be represented using apposition or NPs to the left. These sometimes include the *other* PER entity in the relation, but this is not compulsory.

{Hulk Hogan's son}$_d$, {Nick Bollea}$_e$, ... (apposition)

First Lady Laura Bush and {daughter}$_d$ {Jenna}$_e$ ... (an NP to the left)

Sponsor A person's sponsor is typically an organisation who they are associated with, expressed using a possessive to the left.

(45) {Bear Stearns}$_d$' {Cayne}$_e$ gives up CEO position in latest Wall Street shake-up. (a possessive to the left)

(46) ... trounced {Liou Chen Kuang}$_e$, from the opposition {Democratic Action Party}$_d$, ... (a PP to the right)

(47) By {GRANT PECK}$_e$ {Associated Press}$_d$ Writer (an NP to the right)

Type Finally, people's types or roles are often described locally, as apposition, copula constructions, possessive or unmarked NPs to the left, right or relative clauses.

(48) {A former sports minister}$_d$, {Ndiaye}$_e$ has been... (apposition)

(49) {Rusdan}$_e$ is {an Afghan-trained militant}$_d$ (copula)

{Indian Defense Minister}$_d$ {A.K. Anthony}$_e$ (an NP to the left)

Label	KB count	KB %	NIL count	NIL %
NONE	493	43	411	36
LOC	343	30	141	12
ORG	142	12	188	16
PER	145	12	384	34
TOTAL	1,123	100	1,124	100

Table 7.3: Distribution of local description in TAC 11 queries.

(50) {Michael Smoot}$_e$, {Ph.D.}$_d$ (an NP to the right)

(51) {Adolfo Munoz}$_e$ of Seville's {medical staff}$_d$ (a PP to the right)

(52) ...said {M.A. Mohiuddin}$_e$, {whose textile mill makes goods for export}$_d$. (a relative clause to the right)

7.4 Local description in TAC 11

Having tagged each of the sentences that mention a TAC 11 query, we are able to count how entities are locally described. Table 7.3 shows the distribution of descriptions, in total and by query type.[6] The majority of both KB and NIL queries have some form of local description. Wikipedia's coverage and concept of notability and infobox distribution determine whether a query is classified KB or NIL. Mentions for NIL queries are marginally more often described (8%) and of them, people are most often described followed by organisations and then locations. With KB mentions we see almost the reverse pattern, where locations are most often described, followed by people and organizations.

Table 7.4 shows how the different information is distributed for entity types. The main description for locations is place, although this is overwhelmingly used for KB mentions. Where organisations are described, it is usually an alias. Place and type

[6] We are unable to map five queries to sentences.

Type	Label	KB count	KB %	NIL count	NIL %
NONE		493	41	411	33
LOC	address	0	0	5	0
	desc	4	0	5	0
	isplace	8	0	7	0
	orient	22	1	25	2
	place	291	24	73	6
	type	59	4	65	5
ORG	address	0	0	1	0
	alias	90	7	88	7
	place	22	1	59	4
	sponsor	9	0	13	1
	type	40	3	51	4
PER	age	13	1	25	2
	place	13	1	11	0
	relation	4	0	11	0
	sponsor	1	0	6	0
	type	126	10	354	29
TOTAL		1,195	100	1,210	100

Table 7.4: Distribution of local description by information type

are also described, more so for NIL mentions than KB mentions. People are most often described by their type, again, more so for NIL mentions. We note at this point that as we select subsets using more than two conditions (i.e. type, label and KB/NIL), some sets have very few examples (< 10) and so we cannot draw reliable conclusions from statistics based on these low-count sets.

Table 7.5 shows the detailed distribution for location, showing that place is usually specified using the loc-right-comma structure. Type is also specified, using apposition and NPs to the left.

Acronyms are the main way that organisation aliases are described (see Table 7.6). Types are also specified using a wider range of syntactic forms such as apposition

Type	Realisation	KB count	KB %	NIL count	NIL %
address	loc-right-np	0	0	5	2
desc	appos	4	1	4	2
	relclause	0	0	1	0
isplace	loc-left-comma	4	1	7	3
	org-in-pp	3	0	0	0
	per-appos	1	0	0	0
orient	loc-appos	10	2	13	6
	loc-right-copula	2	0	1	0
	loc-right-np	6	1	8	4
	loc-right-pp	3	0	2	1
	loc-right-relclause	1	0	1	0
place	loc-right-comma	282	73	67	36
	loc-right-np	3	0	0	0
	loc-right-pp	7	1	8	4
type	appos	15	3	16	8
	left-np	37	9	42	22
	right-np	7	1	11	5
TOTAL		385	100	186	100

Table 7.5: Distribution of LOC local description

and NPs to the right and left. A NIL organisation's place is described using NPs to the left, PPs to the right and possessive to the left. The type of mentions is described using apposition, NPs to the left and right.

Finally, Table 7.7 shows how person descriptions are specified. Age is mostly specified using apposition for NIL mentions. A relation apposition (e.g. her husband) is a strong indicator that a mention is NIL as it is rare for both people in a family relationship to be notable. Type is the most common description and using apposition is more common with NIL mentions. Pseudo-titles, or NPs to the left are common type indicators, but apply to both KB and NIL mentions.

Type	Realisation	KB count	KB %	NIL count	NIL %
address	loc-right-np	0	0	1	0
alias	acro	89	54	86	40
	appos	1	0	2	0
place	loc-appos	4	2	7	3
	loc-left-np	6	3	17	7
	loc-left-pos	9	5	15	6
	loc-right-comma	0	0	2	0
	loc-right-np	0	0	1	0
	loc-right-pp	5	3	17	7
sponsor	org-appos	0	0	1	0
	org-left-np	1	0	2	0
	org-left-pos	2	1	3	1
	org-right-pp	1	0	4	1
	per-appos	1	0	0	0
	per-left-pos	3	1	3	1
	per-right-pp	1	0	0	0
type	appos	12	7	21	9
	left-np	11	6	20	9
	right-copula	2	1	1	0
	right-np	15	9	9	4
	right-relclause	1	0	3	1
TOTAL		164	100	215	100

Table 7.6: Distribution of ORG local description

This analysis demonstrates that each type of mention is described in terms of different attributes. The attributes can be realised using different syntactic forms, and the results suggest that there are some useful cues to mention notability to be found in local description.

Type	Realisation	KB count	KB %	NIL count	NIL %
age	appos	10	5	18	4
	left-np	3	1	7	1
place	loc-appos	1	0	1	0
	loc-left-np	1	0	1	0
	loc-left-pos	6	3	2	0
	loc-right-np	0	0	1	0
	loc-right-pp	5	2	6	1
relation	per-appos	2	1	11	2
	per-left-np	2	1	0	0
sponsor	org-left-pos	0	0	3	0
	org-right-np	1	0	2	0
	org-right-pp	0	0	1	0
type	appos	26	14	178	41
	copula	3	1	10	2
	left-np	109	62	159	36
	right-np	0	0	2	0
	right-pp	1	0	16	3
	right-relclause	4	2	14	3
TOTAL		174	100	432	100

Table 7.7: Distribution of PER local description

7.5 Extracting local description

The previous section analyses how a human reader can take advantage of local description to disambiguate mentions. Our automated systems will attempt to replicate this where possible using rules that match syntactic and NE patterns. These are manually developed in the context of the analysis above, but with some important differences. We do not replicate all observed patterns and concentrate on patterns that are simple to express. We do not focus on matching exact description spans, instead restricting matching context to a likely description. The descriptions will thus

be noisy and hard to evaluate directly. Our evaluation instead checks the set of rules that applied for a particular query. Given the set of rules extracted by the system and in the gold standard, Equations 7.1 to 7.3 shows how precision, recall and F are calculated. As we aggregate rules across all sentences that a query mention appears in, there is no guarantee that a rule match is firing in the appropriate sentence (where a gold description is marked) or that it is capturing a description. Our evaluation coarsely indicates how well naïve syntactic rules can extract local description.

$$P = \frac{|rules_{system} \cap rules_{gold}|}{|rules_{system}|} \tag{7.1}$$

$$R = \frac{|rules_{system} \cap rules_{gold}|}{|rules_{gold}|} \tag{7.2}$$

$$F = \frac{2PR}{P+R} \tag{7.3}$$

We re-use the Pattern system from Chapter 6 to extract apposition-based description, mapping them to the appropriate types. To extract other realisations, we use the same framework to specify patterns, including the following symbols:

- **LOC, PER, ORG** Named entities.
- **"," and "NOT-,"** A comma or a token that is not a comma.
- **JJ, NN, DT, NP, POS** An adjective, noun, determiner, noun phrase[7] or possessive marker.
- **LOC-TYPE** A location type: town, city, hamlet, village, island, region, area
- **AGE** An age specifier (e.g. Eight-year-old)
- **RELATION** The relation gazetteer from Chapter 6

The location rules in Figure 7.1 are relatively precise at 74%, but are lower recall for an F of 53%, (see Table 7.8). The most common case of description is loc-place-right-comma, but this can be confused with lists of commas, so we explicitly exclude

[7] As defined in Chapter 6.

loc-isplace/loc-in-comma-np	{LOC}$_d$, {LOC}$_e$
loc-isplace/loc-in-pp	{LOC}$_d$ (in|at) {LOC}$_e$
loc-place/loc-right-comma	NOT-, {LOC}$_e$, {LOC}$_d$
loc-place/loc-right-pp	{LOC}$_e$ in {JJ? LOC}$_d$
loc-type-left-np	{JJ? NN * LOC-TYPE}$_d$ of {LOC}$_e$
loc-type-right-np	{LOC}$_e$ {NP}$_d$

Figure 7.1: Location description rules.

Class	Extracted	Gold	P	R	F
loc-isplace/loc-left-comma	28	11	21	54	30
loc-place/loc-right-comma	130	349	86	32	46
loc-place/loc-right-pp	5	15	80	26	40
loc-type-left-np	49	79	85	53	65
loc-type-right-np	75	18	16	66	25
Total	267	484	74	41	53

Table 7.8: Location description performance on TAC 11.

commas, leading to lower recall. The isplace information is relatively rare and diverse, so we decided not to extract it unless the mention is the place of another location (the most common form).

Figure 7.2 shows organisation description rules and Table 7.9 extraction performance. The most common organisation description is an alternative name specified using an acronym. We already take acronyms into account in coreference resolution and so do not extract them here. The remaining types of description are less prominent, with the most common occurring for just 33 queries. Our rules to extract these more varied descriptions are the noisiest of the mention entity types, with performance measured at 42%.

Finally, Figure 7.3 shows the person description rules, and their extraction performance (see Table 7.10) which are perform the best at 63%. Person type rendered as an NP to the left of the mention is the most frequent description and is relatively

org-place/loc-left-np	{LOC}$_d$ JJ? NN * {ORG}$_e$
org-place/loc-left-pos	{LOC}$_d$ POS {ORG}$_e$
org-place/loc-right-pp	{ORG}$_e$ (in|at|near|of) {LOC}$_d$
org-place/loc-right-np	{ORG}$_e$ {LOC}$_d$
org-sponsor/org-left-np	{ORG}$_d$ {ORG}$_e$
org-sponsor/org-left-pos	{ORG}$_d$ POS {ORG}$_e$
org-sponsor/org-right-pp	{ORG}$_e$,? (from|at|of) DT? JJ? {ORG}$_d$
org-sponsor/per-left-pos	{PER}$_d$ POS {ORG}$_e$
org-sponsor/per-right-pp	{ORG}$_e$ of {PER}$_d$
org-type-left-np	{JJ? NN?}$_d$ {ORG}$_e$
org-type-right-copula	{ORG}$_e$ (is|was) {NP}$_d$
org-type-right-np	{ORG}$_e$ {NP}$_d$

Figure 7.2: Organisation description rules.

Class	Extracted	Gold	P	R	F
org-place/loc-left-np	14	23	57	34	43
org-place/loc-left-pos	13	24	76	41	54
org-place/loc-right-pp	24	22	62	68	65
org-sponsor/org-left-np	2	3	0	0	0
org-sponsor/org-left-pos	10	5	20	40	26
org-sponsor/org-right-pp	6	5	50	60	54
org-sponsor/per-left-pos	5	6	60	50	54
org-sponsor/per-right-pp	1	1	100	100	100
org-type-appos	11	33	18	6	9
org-type-left-np	64	31	28	58	37
org-type-right-copula	9	3	22	66	33
org-type-right-np	182	24	4	37	8
Total	279	330	45	38	42

Table 7.9: Organisation description performance on TAC 11.

well extracted at 73%. Type expressed using an apposition is much less successfully extracted at only around 10% recall, so while it is efficient, the Pattern system may not be adequate for this extraction.

per-age-left-np	{AGE}$_d$ {PER}$_e$
per-place/loc-left-pos	{LOC}$_d$ POS NP? TITLE {PER}$_e$
per-place/loc-right-pp	{PER}$_e$ (from|of) {LOC}$_d$
per-relation/per-left-np	{PER POS RELATION +}$_d$,? {PER}$_e$
per-relation/per-right-np	{PER}$_e$,? {RELATION + (of PER)}$_d$
per-relation/per-right-relclause	{PER}$_e$,? {whose RELATION + PER}$_d$
per-sponsor/org-left-pos	{ORG}$_d$ POS {PER}$_e$
per-sponsor/org-right-pp	{PER}$_e$,? (in|of|at|from) DT? NN? {ORG}$_d$
per-type-left-np	{NP}$_d$ {PER}$_e$
per-type-right-relclause	{PER}$_e$, {who TOKEN+}$_d$,

Figure 7.3: Person description extraction performance and rules.

Class	Extracted	Gold	P	R	F
per-age-appos	24	28	75	64	69
per-age-left-np	10	10	80	80	80
per-place/loc-left-pos	11	8	54	75	63
per-place/loc-right-pp	3	11	66	18	28
per-relation/per-appos	11	13	72	61	66
per-relation/per-left-np	16	2	12	100	22
per-relation/per-right-np	2	0	0	0	0
per-sponsor/org-left-pos	5	3	60	100	75
per-sponsor/org-right-pp	10	1	0	0	0
per-type-appos	79	204	25	9	14
per-type-left-np	255	268	75	71	73
per-type-right-relclause	34	18	29	55	38
Total	322	529	83	51	63

Table 7.10: Person description performance on TAC 11.

We followed the methodology for pattern creation as explained in Chapter 6. The statistics in Section 7.4 informed pattern design, but we did not undergo multiple iterations as we did in apposition extraction. This is because we cannot directly evaluate extraction performance without token-level annotation and also because we

focus on linking performance. More accurate extraction would benefit linking and future work will involve more sophisticated methods.

7.6 Linking with local description

We create several features for our supervised models ($\mathcal{S}$, $\mathcal{S}+$) that depend on local descriptions. Our features model where in the candidate article different types of description match. We convert each description into a bag of words, removing stopwords and expanding demonym forms of countries or abbreviations of states. Where we find an age, and the document has a timestamp, we convert it to the feasible year of birth, assuming that the age was true at the document timestamp. We also filter some personal titles (e.g. Mr, Mrs, Miss and Madame). We extract fields from the candidate article: first sentence, first paragraph, first section, section title, title, categories, infobox values and tokens. These are converted to bags of words and we check to see if they contain unigrams and bigrams from each of the descriptions. We generate features that model which descriptions matched and to which field.

Table 7.11 shows the features generated when a org-sponsor/per-left-pos rule extracts a description whose unigrams are found in a candidate tokens field. These include summary features describing that descriptions were found with at least one matching some field (i.e. *Any*) and that a specific rule fired. Where a match occurs, we generate summary features that show that the rule has matched, that the tokens field has matched and the unigram overlap between them. Note that this example's description is itself a unigram and the bigram matching feature does not apply. For any fields that do not match, we generate a field summary feature which lists which fields didn't match a rule, and combinations of rules that fired and the field it failed to match—explicitly modelling a locally described entity that is not found in the KB.

Table 7.12 shows the features generated where the same rule fires, but does not match any fields and would include the no match features from Table 7.11 and the corresponding feature for the tokens field.

Type	Description	Field	Value
Summary	*Any*	*Any*	1
	org-sponsor/per-left-pos		1
Match	org-sponsor/per-left-pos	*Any*	1
	Any	`tokens`	1
	org-sponsor/per-left-pos	`tokens-ug`	†1.0
No match	*Any*	`categories, first_paragraph, first_section, first_sentence, infobox, section_titles, title`	1
	org-sponsor/per-left-pos	`categories, first_paragraph, first_section, first_sentence, infobox, section_titles, title`	1

Table 7.11: Features generated from an org-sponsor/per-left-pos description matching an article's tokens. Values are binary, except for overlap values (†).

Type	Description	Field	Value
Summary	*Any*	*None*	1
	org-sponsor/per-left-pos		1

Table 7.12: Features generated from an org-sponsor/per-left-pos description without a match. No match fields are as above, but adding features for the token field.

These features are used in the supervised model and are also used in clustering to split NIL clusters where queries have contradictory attributes. More specifically, we apply the rule-based clustering method using the attributes: W, KB, m_l. This clusters queries by Wikipedia article, or the same TAC KB entry or longest mention if no link was assigned. Then, we examine all clusters that are linked to NIL and share m_l. If all chains have the local type attribute extracted from any of their mentions and we can

partition the cluster in two, then we split the cluster by the local description attribute. We ignore cases which do not fit that pattern and do not consider splitting clusters into more than two subsets. This approach is consistent with our attribute-based rules, though these features could inform a more sophisticated clustering method.

The extraction statistics above illustrate how local description can be extracted from TAC 11 documents. Table 7.13 shows how the candidates for TAC 11 queries match Wikipedia articles. The match columns show how many candidates match of the LINK and NOLINK classes. Ideally, we would see no matches from candidates labelled NOLINK. All rules have some amount of noise and we see more NOLINK matches than LINK matches in every class. We do not restrict most of our rules for semantic type and this is a major source of noise. For example, our most-extracted class of descriptions is per-type, and many of them captured by an NP to the left of the mention. Without restricting these so that a description contains a valid attribute for a person (e.g. ensuring that the description is a role) there is a low precision 7:1 ratio of NOLINK to LINK matches.

The no match columns show how many candidates with the different labels do not match. In this case, an ideal case would be that we find no LINK candidates and many NOLINK candidates that fail to match. The ratio of NOLINK to LINK is more favourable here.[8] Improvement here may require capturing more description context in the document or generalising descriptions. Currently, director will not match filmmaker and some generalisation of these descriptions is necessary to match the two.

Table 7.14 shows which Wikipedia article fields are matched by different rules, counting the number of candidates. There is some difference in which fields match, but we note that this table does not discriminate between correct and incorrect matches. The tokens field always matches the most candidates because it contains most of the other fields except categories and infoboxes. Interestingly, we see that title matches are fairly frequent for some rules: loc-place, org-type and per-type. This means that the extracted description is matching, presumably in content after

[8] And in the right direction.

Type	Label	Match		No match	
		LINK	NOLINK	LINK	NOLINK
LOC	isplace	2	36	3	134
	place	97	179	2	364
	type	47	233	14	478
ORG	place	15	33	1	143
	sponsor	7	15	0	151
	type	73	297	33	804
PER	age	9	23	1	220
	place	5	8	0	6
	rel	0	0	3	34
	relation	0	0	4	49
	sponsor	4	17	0	77
	type	97	704	21	2042

Table 7.13: TAC 11 query matches. We show the number of matching candidates labelled LINK and NOLINK, and the number of non-matching LINK and NOLINK candidates.

a comma (e.g. `Melbourne, Ontario`) or in parentheses (e.g. `John Howard (Australian actor)`). The count increases as the match can include the first sentence (Sent), paragraph (Para) and section (Sect). Section titles (S. Titles) match some content, but much less than infoboxes (Info) and categories (Cat). We will analyze the impact of this matching via linking performance below, but the field count differences are substantial. This suggests that there is some difference, if not value, in viewing Wikipedia articles as a more structured source of information than simply a bag of words.

7.6.1 Results

We have analysed how entity mentions are described, proposed rules to restrict context to descriptions and modelled them as features. Table 7.15 shows performance on TAC 11 and TAC 12 data for the best systems from the literature and our linkers

Type	Label	Title	Sent	Para	Sect	S. Titles	Info	Cat	Tokens
LOC	isplace	1	4	6	9	1	21	9	36
	place	106	210	229	209	9	196	196	298
	type	26	109	155	158	28	110	60	282
ORG	place	24	39	41	37	7	39	30	48
	sponsor	3	6	13	13	2	16	5	18
	type	59	103	153	163	80	135	76	389
PER	age	0	13	13	13	0	13	0	30
	place	1	12	12	10	1	12	11	13
	rel	0	0	0	0	0	0	0	0
	relation	0	0	0	0	0	0	0	0
	sponsor	0	2	9	9	1	8	8	22
	type	83	243	327	344	81	316	195	936

Table 7.14: TAC 11 field matches. We show the number of candidates that match in each of the Wikipedia article fields (with the most specific at the left).

that model local description. The $\mathcal{S}$ system is trained on TAC 09, TAC 10 and $\mathcal{S}+$ uses TAC 11 as well.

Adding local description features to $\mathcal{S}$ leads to minor improvements. On TAC 11, the accuracy increases 0.2% to 89.4% and B^{3+} increases 0.1% to 86.4. This is driven by increases in KB performance: 0.4% accuracy and 0.6% B^{3+}. We see minor gains in TAC 12: a 0.1% increase in accuracy and a 0.2% increase in B^{3+}. This is due to an increase in performance over NIL queries: 0.9% accuracy and 0.9% in B^3 F. The effect is similar for $\mathcal{S}+$, with NIL performance increasing more than the KB equivalent declines. Using local description to split NIL clusters has a small effect. Performance decreases slightly on TAC 11 NIL clustering, but has no impact overall. We see a 0.3% increase in B^{3+} NIL F in TAC 12, which translates to a 0.1% increase in B^{3+} F.

Our results suggest the utility of local features for our NEL system and similar features have also been used to good effect elsewhere. Cucerzan and Sil (2013) also use local features, but in a framework that does not assume one sense per discourse.

System	Dataset	Accuracy			B^{3+} F		
		All	KB	NIL	All	KB	NIL
Monahan et al. (2011) (TAC)	11	86.1	-	-	84.6	-	-
Zhang et al. (2012)	11	87.6	-	-	-	-	-
$\mathcal{S}$	11	89.2	83.1	95.4	86.3	81.3	91.2
+ local	11	**89.4**	**83.5**	95.2	**86.4**	**81.9**	91.0
+ local + clust	11	**89.4**	**83.5**	95.2	**86.4**	**81.9**	90.9
Cucerzan (2012) (TAC)	12	76.6	-	-	73.0	68.5	78.1
$\mathcal{S}$	12	74.7	66.2	84.2	70.4	62.9	78.9
+ local	12	**74.8**	65.7	**85.1**	**70.6**	62.4	**79.8**
+ local + clust	12	**74.8**	65.7	**85.1**	**70.7**	62.4	**80.0**
$\mathcal{S}+$	12	75.8	69.2	83.3	71.5	65.7	78.0
+ local	12	**76.0**	68.6	**84.4**	**71.7**	65.1	**79.1**
+ local + clust	12	**76.0**	68.6	**84.4**	**71.8**	65.1	**79.4**

Table 7.15: Local description linking performance.

An analysis of the separate impact of their local features has not been made public. However their system shows similar performance with overall accuracies of 89.9% on TAC 11 and 80.4% on TAC 12 (Silviu Cucerzan, PC, 2014).

This chapter attempts to model mention descriptions in such a way that they can help disambiguation. We performed initial ablative analysis using the system reported above, but removing features at that threshold resulted in the same or better performance. Departing from our initial threshold, we optimise our threshold using the TAC 11 data, having trained $\mathcal{S}$ on TAC datasets from 2009 and 2010 and $\mathcal{S}+$ on 2009, 2010 and 2011. The threshold value 0.1 is used for all three systems and the results presented in Table 7.16. Again, local features help linking with improvements in accuracy and B^{3+} F overall.

We use a permutation test (Chinchor, 1995) to check whether the difference between $\mathcal{S}$ and $\mathcal{S}$ with local features is statistically significant. We run 10,000 trials where the system responses are randomly swapped, and measure the difference between

System	Dataset	Accuracy			B^{3+} F		
		All	KB	NIL	All	KB	NIL
Cucerzan (2012) (TAC)	12	76.6	-	-	73.0	68.5	78.1
$\mathcal{S}$ (t=0.1)	12	74.8	69.7	80.7	70.4	65.7	75.6
(t=0.1) + local	12	⋆**75.5**	69.7	**82.0**	71.2	**66.0**	77.0
(t=0.1) + local + clust	12	⋆**75.5**	69.7	**82.0**	**71.3**	**66.0**	**77.2**
$\mathcal{S}+$ (t=0.1)	12	75.4	71.6	79.7	71.1	67.8	74.8
(t=0.1) + local	12	**76.1**	68.7	**84.4**	71.8	65.2	79.1
(t=0.1) + local + clust	12	**76.1**	68.7	**84.4**	**71.9**	65.2	**79.4**

Table 7.16: Linking performance using TAC 11 threshold (t=0.1). ⋆ indicates statistical significance at $p < 0.05$. Bold indicates the best performance of a system on a dataset for a particular metric.

their accuracies. The p-value is the fraction of trials with a difference greater than equal to the original difference in accuracies. As we cannot guarantee that our NIL IDS are stable between systems (i.e. NIL001 may not mean the same thing in two outputs, where EL001 will), we do not assess NIL clustering metrics. We find that local features make a statistically significant difference at p=0.04 for $\mathcal{S}$ with and without clustering, but no significant difference for $\mathcal{S}+$ (p=0.11).

Table 7.17 shows the impact of different local features on accuracy. Location place, organisation sponsors, and personal relations and sponsors are the most useful descriptions for disambiguation. These are the most obviously flagged descriptions, using commas or apostrophes. Person ages, places and types have some positive impact. Type features in general have less positive impact, or in the case of locations, no impact. Personal types are interesting, as they are the most frequent type of personal description (see Table 7.4), but perhaps the variable extraction performance (14.1% to 73.4% F in Table 7.3) is too noisy for the feature to have significant advantages. Removing organisation place descriptions increases performance by -0.1% accuracy, so perhaps these are not adequately modelled.

Linker	12
$\mathbb{S}$	75.5
-loc-isplace	-0.2
-loc-place	-0.2
-loc-type	0.0
-org-place	+0.1
-org-sponsor	-0.2
-org-type	-0.1
-per-age	-0.1
-per-place	-0.1
-per-rel	-0.2
-per-sponsor	-0.2
-per-type	-0.1

Table 7.17: Ablative analysis for $\mathbb{S}$ using the threshold optimised on TAC 11.

7.6.2 Feature weight analysis

An advantage of using log linear models is that the weights can be examined.[9] Table 7.18 shows a sample of feature weights from $\mathbb{S}$, where positive weights indicate good evidence for linking to a KB candidate. Context similarity and KB statistics are important: `unigram_cosine` is first, followed by `kb_refprob` with respect to the candidate list and for a candidate. Matching the structure of a KB is important as well, with `cat_score` and `context_score` highly ranked. Local features that indicate a match are when an organisation type specified as an NP matched a candidate infobox field (e.g. ... investment bank ABN...), or if a location is specified as a place and matches a candidate's first paragraph. For example, the mention Florida may be the place description for Atlanta (e.g. Atlanta, Florida). Thus in the candidate `Florida`, we might expect to see Atlanta mentioned in the first paragraph.

[9] Note that this analysis does not make assumptions about feature independence, but is simply the final state of the model.

Feature	Weight
unigram_cosine_similarity	0.479
ccs_kb_refprob_irank	0.341
kb_refprob	0.338
ccs_kb_search_irank	0.273
ccs_unigram_cosine_similarity_irank	0.269
ccs_cat_score_irank	0.262
bigram_cosine_similarity	0.261
local-m-org-type-left-np-infobox-ug	0.252
title_dice	0.244
local-m-loc-isplace/loc-in-comma-np-first_paragraph-ug	0.241

Table 7.18: Top 10 positive weights for $\mathcal{S}$.

Negative weight indicates that a feature does not indicate a link and low scores for all mention candidates will lead to a NIL link in our models. Table 7.19 shows that local features have more impact as negative evidence, with the absence of local contexts the strongest. Our analysis above suggests that KB and NIL mentions are described in similar proportions, but perhaps our low recall extraction extracts descriptions for NIL mentions less often such that no description indicates a NIL. Finding description that does not match is also good NIL evidence, in general, but also some specific cases such as: organisation places in section titles, organisation types in infoboxes, as NP or apposition. Other prominent negative weights are entity type mismatches, but there are some interesting negatively weighted features. The best match between an organisation type and the tokens is weighted negatively, as are any match against the tokens in the document, the presence of any person type or organisation place (not shown but both are -0.16). These indicate that comparing descriptions against the tokens results in spurious matches, so even if a description validly matches the candidate article text, it may be penalised.

Finally, we revisit the whole document task setting introduced in Section 5.3.5 and evaluate linkers with local description on the TEST section of the SMH corpus. Table

Feature	Weight
local-m-None-tokens	-0.165
lORG-MISC	-0.173
l?-MISC	-0.181
local-nm-H(ORG|GPE)_break_A(DT_NP)_end-infobox	-0.192
local-nm-org-type-left-np-infobox	-0.198
ccs_local-m-org-type-left-np-tokens-ug_irank	-0.216
local-nm-org-place/loc-right-pp-section_titles	-0.232
local-found-nm	-0.266
ccs_kb_chain_refprob_irank	-0.346
local-nofound-nm	-0.650

Table 7.19: Top 10 negative weights for $\mathcal{S}$.

System	All	KB	NIL
$\mathcal{U}$ (t=0.25)	71.3	72.9	66.7
$\mathcal{S}$ (t=0.1)	70.6	72.1	65.8
$\mathcal{S}+$ (t=0.1)	70.7	72.3	65.9
$\mathcal{S}$SMH (t=0.2)	71.7	74.2	65.1
$\mathcal{S}+$ +local (t=0.05)	70.2	71.5	68.8
$\mathcal{S}$SMH +local (t=0.15)	72.6	74.6	67.1

Table 7.20: Results on SMH TEST. Thresholds are optimised on SMH DEV.

7.20 shows the results, repeating for comparison our original systems from Table 5.22. Local features decrease performance on the TAC-trained supervised system on SMH data by 0.5% at 70.2% F overall. We see a 0.9% increase on the SMH trained model to 72.6%, our highest on the SMH TEST dataset. We use another permutation test to show that the difference between the SMH models, with and without local features, is statistically significant at p=0.0006 over 10,000 trials. This result shows that local features increase linking performance outside the TAC task context.

7.7 Summary

This chapter has explored how entity mentions are described and how this information can be used for linking to a KB, broadening out the narrow class of descriptions represented using appositions in the previous chapter. While it results in performance improvements, more work is required to take full advantage of local description.

Extraction is the first challenge. Our extraction rules capture largely nominal patterns and are, for the most part, semantically naïve. As the descriptions are fundamental to the matching, more sophisticated extraction would be beneficial, including integrating full relation extraction systems, syntactic parsing to recover verb-mediated description (e.g. John Howard was appointed Prime Minister...) (Yao et al., 2013; Ling et al., 2014) and inducing richer semantic types for mentions (Nakashole et al., 2013; Ling and Weld, 2012; Lin et al., 2012b).

Matching can be difficult when descriptions are sparse. For example, a human annotator may reason that a filmmaker and director are the same role, whereas our system may not. Integrating with WordNet and other semantic resources like YAGO would help generalise semantic types. Descriptions may be misleading where there is a mistake, for example the typographical error confusing countries in the sponsor description: ...{Austria}$_d$'s {AMCI}$_e$ Holdings. Moreover, temporal issues can be problematic, as people hold different positions over a career and arbitrary splits might consider Arnold Schwarzenegger the politician as different to the actor. Our current system only minimally processes the Wikipedia article structure, separating it into different fields. Our feature weight analysis suggests that there is some advantage to targeted matching (e.g. to infobox fields) where token matching can be counterproductive. While extracting description from the articles is an obvious extension, even understanding how different phases of a person's life are expressed in an article may help. Capturing dependencies between different attributes as in Garera and Yarowsky (2009) would help build more comprehensive profiles. Our clustering

approaches are naïve by design and less sparse descriptions and more sophisticated clustering (Mann and Yarowsky, 2003) would likely help.

In conclusion, local description can help disambiguate ambiguous names. Our system attempts to use these attributes to model matching candidate KB entries and the absence of a match. This improves our state-of-the-art results to within 0.5% accuracy and 1.1% B^{3+} F over all clusters of the top system in TAC 12.

8 Conclusion

```
John Smith (cricketer, born 1833)
John Smith (cricketer, born 1834)
John Smith (cricketer, born 1835)
John Smith (cricketer, born 1841)[1]
John Smith (cricketer, born 1843)
```

Five John Smith cricketers born only a decade apart.

Name ambiguity is a major challenge in understanding natural language. This thesis has explored how to resolve name ambiguity by linking name mentions to a knowledge base. This task requires identifying the correct KB entry for a mention in text (a KB link) or NIL if the mention refers to an entity outside the KB. Chapter 2 reviewed work from coreference resolution, within a document and across a corpus, establishing contextual matching as a key technique. Wikification poses the problem of linking text to a KB, using statistics and semantic structure derived from it. When an entity is mentioned in a discourse, it is often introduced or accompanied by descriptions and while these attributes may match a KB entry's broad content, existing approaches have not directly attempted to use description for disambiguation.

We touched on datasets and metrics for evaluation in Chapter 3. A diverse range of datasets makes it difficult to compare systems. Our first contribution is formalising a framework for NEL, consisting of extraction, search and disambiguation. This is applied to three seminal systems, which are evaluated on a common dataset in

[1] This title is actually `John Smith (Derbyshire cricketer)`, but is is changed for consistency.

Chapter 4. Our second contribution is the identification and combination of these critical elements of a NEL system.

The Text Analysis Conference is a major venue for NEL research. We participated in their shared task in 2010, 2011 and 2012. Chapter 5 described the principles that inform our system design and how the system evolved to incorporate prominent ideas from the literature. Our third contribution is our state-of-the-art linker and its thorough analysis, that is, in some sense, a meta-system, combining some of the best ideas from the literature.

In the remainder of the thesis, we sought to characterise entity description. Chapter 6 examined apposition, a prominent way of describing entities. We modelled the syntactic and semantic restrictions from the linguistic theory and used a joint classifier to extract apposition from a multi-genre corpus. Our apposition extraction system improved on the state of the art and is our fourth contribution.

While apposition is useful, it is only one of the ways that entities are described. Chapter 7 analysed *local description*, where attributes are specified close to the mention they describe. Our analysis of the TAC 11 queries shows that, from a human annotator's point of view, there is an observable difference between the way that notable entities and non-notable entities are described. Our rule-based system extracts local description and we represent both context matches and absence of matches with candidate KB entries. Our final contributions are the cataloguing and analysis of local description for TAC queries, and demonstrating that local features improve the performance of our supervised system, especially identifying NIL mentions.

8.1 Future work

There is substantial existing work on NEL, but many more problems to explore. Our work in linking is focused on the TAC task and its evaluation methods and data. We train the systems we evaluate on TAC only on the queries provided, but have not explored other sources. There is a wide range of training data available: TAC

queries, internal Wikipedia hyperlinks (Zhang et al., 2011a,c), links from outside Wikipedia (e.g. the Wikilinks corpus of Singh et al. (2012)), automatically-linked data (Gabrilovich et al., 2013) and a wealth of manually-linked datasets (Cucerzan, 2007; Fader et al., 2009; Kulkarni et al., 2009; Hoffart et al., 2011). Our experiments adapting the TAC system to the SMH news corpus in Chapters 5 and 7 touched on this, moving from an international to an Australian-specific news context.

As we note in Chapter 2, researchers have begun to link directly to other KBs such as Freebase (Zheng et al., 2012; Gabrilovich et al., 2013). Work on local entity descriptions and extracting entity relations from text (Cheng and Roth, 2013) means that linking becomes more similar to entity resolution and record linkage, where structured records are matched together. Since these areas have an established background in handling structural ambiguity at scale, combining elements from unstructured and structured data linkage could benefit solutions to both problems.

Recent TAC evaluations have emphasised NILs and this is another critical aspect of adapting linkers to new domains. The domains usually studied for linking contain mentions to notable entities with Wikipedia articles. This is not the case for other domains and presents a challenge. For example, we may wish to link corporate documents that discuss many entities, few of them notable. Our KB may be a list of employees rather than Wikipedia. Many of the TAC systems have used Wikipedia, with and without textual content, but there has been little exploration of linking to a small custom KB, although the cold-start knowledge base population tasks hint at how this may be done. Handling a knowledge-poor KB is an important factor to consider when linking in a real-world context. The TAC entity linking evaluation assumes that the KB is stable, which is emphatically not the case in the real world. Many systems, including ours, map between the TAC KB and a larger Wikipedia snapshot to optimise for TAC performance. While this detracts from task realism, finding an effective mapping as Wikipedia diverges is essentially the same temporal shift problem that is key to linking against a real, dynamic KB.

There are several interesting directions in which we may improve our own systems and resources. Our experiments have shown that extraction can help identify a better-specified alias of a mention for more accurate search. Improved NER and coreference would improve this further. Typed gazetteers are common in NER systems (Ratinov and Roth, 2009), but the distributions over the types of candidates for a name could be used to hypothesise the types. It may suffice to use a precise, lightweight linker as in Chapter 4 and vote on the types of the candidates returned (e.g. all John Howard candidates are people). Recent work on coreference resolution and linking (Zheng et al., 2013; Hajishirzi et al., 2013) shows the value of a more holistic treatment of the problem of name ambiguity.

Chapter 4 establishes the importance of search for linking and our systems in Chapter 5 use high-recall search. Formulating a query that can precisely characterise an entity remains a challenge. Motivated by work in Pilz and Paaß (2012) that searches for multiple entities at the same time, we may consider adding more disambiguating information to our search: locally described attributes, related concepts or the topic classification of a document. While our parse-based apposition extraction systems are state of the art, we do not use them to extract local description as we wish to avoid a full syntactic parse.

Extracting apposition from noun phrase chunks is an interesting direction, however using a state-of-the-art coreference resolution system may require parsing anyway. More broadly, the rules we use to extract local description are sufficient to provide some benefit to linking, but could be improved perhaps by bootstrapping as in Mann and Yarowsky (2003). Some entities may provide better context than others, for example, mention of a topic: politics or entertainment may help decide between `John Howard` and `John Howard (Australian actor)`. Automatically deciding which evidence is best for disambiguation (Li et al., 2013) is a promising direction, but using this information for search may have efficiency benefits from a smaller, more accurate candidate list.

Finally, our disambiguation and NIL clustering components raise important research questions. Linear models are an efficient and simple way to model linking, but present some issues. Features that encode how well a mention matches a KB entry name could be confounded by ambiguous examples. For example, both `John Howard` and `John Howard` should match the mention John Howard well, but one may be labelled LINK and the other NOLINK. Name match is a precondition for linking, but this situation may result in it being considered negative evidence. Models that better encode feature dependence or rank instances may handle these situations more elegantly. Identifying and clustering NIL mentions remains a significant challenge. Our threshold approach to classification works well, and local description can help identify NILs, but it can be difficult to separate mentions of a minor KB entry (with a very short article) from NIL mentions. Clustering mentions has long been studied in CDCR, but the top-performing systems in TAC have tended to take a "link first, cluster later" approach. The requirement for scalability and interpreting limited context are attractive qualities in CDCR approaches and linking would certainly benefit from a tighter integration.

8.2 Summary

In sum, this thesis has investigated how we can resolve ambiguous textual references by grounding them to a knowledge base. We introduced a framework for analysing linking systems to better understand them. We described in detail our submissions to three years of the TAC shared task and the state-of-the-art linking system that is the result of our participation. We explored apposition extraction and presented a state-of-the-art system that takes advantage of syntactic and semantic aspects of apposition. Finally, we generalised this to analyse, extract and model local description for linking, leading to statistically significant improvements over our state-of-the-art linking systems.

Context is an important cue that we, as readers, use to resolve ambiguity—we should strive to design systems that do the same. While precise local information helps linking, there is much work to be done to create rich entity descriptions that help bridge the gap between the loosely structured text and structured knowledge bases. The framework and techniques described in this thesis are a strong foundation for anyone wishing to link mentions in text to knowledge bases.

Bibliography

2009. *Proceedings of the Text Analysis Conference 2009*. National Institute of Standards and Technology, Gaithersburg, MD USA.

2010. *Proceedings of the Text Analysis Conference 2010*. National Institute of Standards and Technology, Gaithersburg, MD USA.

2011. *Proceedings of the Text Analysis Conference 2011*. National Institute of Standards and Technology, Gaithersburg, MD USA.

2012. *Proceedings of the Text Analysis Conference 2012*. National Institute of Standards and Technology, Gaithersburg, MD USA.

2013. *Proceedings of the Text Analysis Conference 2013*. National Institute of Standards and Technology, Gaithersburg, MD USA.

Eneko Agirre, Angel X. Chang, Daniel S. Jurafsky, Christopher D. Manning, Valentin I. Spitkovsky, and Eric Yeh. 2009. Stanford-UBC at TAC-KBP. In TAC (2009).

Ivo Anastácio, Bruno Martins, and Pável Calado. 2011. Supervised learning for linking named entities to knowledge base entries. In TAC (2011).

Nicholas Andrews, Jason Eisner, and Mark Dredze. 2012. Name phylogeny: A generative model of string variation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 344–355.

Alan R. Aronson. 2001. Effective mapping of biomedical text to the UMLS metathesaurus: the metamap program. *Proceedings of the American Medical Informatics Association Symposium*, pages 17–21.

Javier Artiles, Julio Gonzalo, and Satoshi Sekine. 2007. The SemEval 2007 WePS Evaluation: Establishing a benchmark for the Web People Search task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 64–69.

Javier Artiles, Julio Gonzalo, and Satoshi Sekine. 2009. WePS 2 evaluation campaign: Overview of the Web People Search clustering task. In *Proceedings of the WWW Web People Search Evaluation Workshop*.

Javier Artiles, Qi Li, Taylor Cassidy, Suzanne Tamang, and Heng Ji. 2011. CUNY BLENDER TAC-KBP2011 temporal slot filling system description. In TAC (2011).

Amit Bagga and Breck Baldwin. 1998a. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 560–567.

Amit Bagga and Breck Baldwin. 1998b. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 79–85.

BBN. 2004–2007. Co-reference guidelines for English OntoNotes. Technical Report v6.0, BBN Technologies.

Eric Bengtson and Dan Roth. 2008. Understanding the value of features for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 294–303.

Luisa Bentivogli, Pamela Forner, Claudio Giuliano, Alessandro Marchetti, Emanuele Pianta, and Kateryna Tymoshenko. 2010. Extending English ACE 2005 corpus

annotation with ground-truth links to Wikipedia. In *Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 19–27.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005.

Indrajit Bhattacharya and Lise Getoor. 2007. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data*, 1(1).

Indrajit Bhattacharya, Shantanu Godbole, and Sachindra Joshi. 2008. Structured entity identification and document categorization: two tasks with one joint model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 25–33.

Dan Bikel, Vittorio Castelli, Radu Florian, and Ding-jung Han. 2009. Entity linking and slot filling through statistical processing and inference rules. In TAC (2009).

Roi Blanco and Hugo Zaragoza. 2010. Finding support sentences for entities. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 339–346.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Ludovic Bonnefoy and Patrice Bellot. 2012. LIA at TAC KBP 2012 English entity linking track. In TAC (2012).

Elizabeth Boschee, Ralph Weischedel, and Alex Zamanian. 2005. Automatic information extraction. In *Proceedings of the Conference on Intelligence Analysis*.

Sergey Brin and Larry Page. 1998. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International World Wide Web Conference*, pages 107–117.

Razvan Bunescu and Marius Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 9–16.

Elena Cabrio and Bernardo Magnini. 2010. Toward qualitative evaluation of textual entailment systems. In *Coling 2010: Posters*, pages 99–107.

Arnaldo Candido, Erick Maziero, Lucia Specia, Caroline Gasperin, Thiago Pardo, and Sandra Aluisio. 2009. Supporting the adaptation of texts for poor literacy readers: a text simplification editor for brazilian portuguese. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 34–42.

Yunbao Cao, Chin-Yew Lin, and Guoquing Zheng. 2011. MSRA at TAC 2011: Entity linking. In TAC (2011).

Jean Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22:249–254.

Taylor Cassidy, Zheng Chen, Javier Artiles, Heng Ji, Hongbo Deng, Lev-Arie Ratinov, Jing Zheng, Jiawei Han, and Dan Roth. 2011. CUNY-UIUC-SRI TAC-KBP2011 Entity Linking System Description. In TAC (2011).

Angel X. Chang, Valentin I. Spitkovsky, Eric Yeh, Eneko Agirre, and Christopher D. Manning. 2010. Stanford-UBC Entity Linking at TAC-KBP. In TAC (2010).

Kai-Wei Chang, Rajhans Samdani, and Dan Roth. 2013. A constrained latent variable model for coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 601–612.

Zheng Chen and Heng Ji. 2011. Collaborative ranking: A case study on entity linking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 771–781.

Zheng Chen, Suzanne Tamang, Adam Lee, Xiang Li, Wen-Pin Lin, Matthew Snover, Javier Artiles, Marissa Passantino, and Heng Ji. 2010. CUNY-BLENDER TAC-KBP2010 entity linking and slot filling system description. In TAC (2010).

Xiao Cheng, Bingling Chen, Rajhans Samdani, Kai-Wei Chang, Zhiye Fei, Mark Sammons, John Wieting, Subhro Roy, Chizheng Wang, and Dan Roth. 2013. Illinois Cognitive Computation Group UI-CCG TAC 2013 entity linking and slot filler validation systems. In TAC (2013).

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796.

Nancy Chinchor. 1995. Statistical significance of MUC-6 results. In *MUC6 '95: Proceedings of the 6th conference on Message understanding*, pages 39–43.

Rudi L. Cilibrasi and Paul M. B. Vitanyi. 2007. The Google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383.

James Clarke, Yuval Merhav, Ghalib Suleiman, Shuai Zheng, and David Murgatroyd. 2012. Basis Technology at TAC 2012 entity linking. In *Proceedings of the Text Analysis Conference, 2012*.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. 2013. A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd international conference on World Wide Web*, WWW '13, pages 249–260.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716.

Silviu Cucerzan. 2011. TAC entity linking by performing full-document entity extraction and disambiguation. In TAC (2011).

Silviu Cucerzan. 2012. MSR system for entity linking at TAC 2012. In TAC (2012).

Silviu Cucerzan and Avirup Sil. 2013. The MSR systems for entity linking and temporal slot filling at TAC 2013. In TAC (2013).

Aron Culotta, Michael Wick, and Andrew McCallum. 2007. First-order probabilistic models for coreference resolution. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 81–88.

James R. Curran and Stephen Clark. 2003. Language independent NER using a maximum entropy tagger. In *Proceedings of the Seventh Conference on Natural Language Learning*, pages 164–167.

Hal Daumé III. 2004. Notes on CG and LM-BFGS optimization of logistic regression. Unpublished.

Alexandre Davis, Adriano Veloso, Altigran Soares, Alberto Laender, and Wagner Meira Jr. 2012. Named entity disambiguation in streaming data. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 815–824.

Tim Dawborn and James R. Curran. 2014. docrep: A lightweight and efficient document representation framework. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 762–771.

David Day, Janet Hitzeman, Michael Wick, Keith Crouch, and Massimo Poesio. 2008. A corpus for cross-document co-reference. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, pages 23–31.

Tom De Smedt and Walter Daelemans. 2012. Pattern for Python. *Journal of Machine Learning Research*, 13:2013–2035.

Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2012. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 469–478.

Sebastian Deorowicz and Marcin G. Ciura. 2005. Correcting spelling errors by modeling their causes. *International Journal of Applied Mathematics and Computer Science*, 15:275–285.

Laura Dietz and Jeffery Dalton. 2012. Across-document neighborhood expansion: UMass at TAC KBP 2012 entity linking. In TAC (2012).

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 277–285.

Bradley Efron. 1979. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1):1–26.

Joe Ellis, Xuansong Li, Kira Griffitt, Stephanie M. Strassel, and Jonathan Wright. 2012. Linguistic resources for 2012 knowledge base population evaluations linguistic data consortium. In TAC (2012).

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2009. Scaling Wikipedia-based named entity disambiguation to arbitrary web text. In *Proceedings of the WikiAI 09 - IJCAI Workshop: User Contributed Knowledge and Artificial Intelligence: An Evolving Synergy*.

Angela Fahrni, Thierry Göckel, and Michael Strube. 2012. HITS' monolingual and cross-lingual entity linking system at TAC 2012: A joint approach. In TAC (2012).

Angela Fahrni, Benjamin Heinzerling, Thierry Göckel, and Michael Strube. 2013. HITS' monolingual and cross-lingual entity linking system at TAC 2013. In TAC (2013).

Angela Fahrni, Vivi Nastase, and Michael Strube. 2011. HITS' cross-lingual entity linking system at TAC 2011: One model for all languages. In TAC (2011).

Angela Fahrni and Michael Strube. 2012. Jointly disambiguating and clustering concepts and entities with Markov logic. In *Proceedings of COLING 2012*, pages 815–832.

Benoit Favre and Dilek Hakkani-Tür. 2009. Phrase and word level strategies for detecting appositions in speech. In *Proceedings of Interspeech 2009*, pages 2711–2714.

Brent D. Fegley and Vetle I. Torvik. 2013. Has large-scale named-entity network analysis been resting on a flawed assumption? *PLoS ONE*, 8(7):e70299.

Ivan P. Fellegi and Alan B. Sunter. 1969. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210.

Norberto Fernández, José M. Blázquez, Luis Sánchez, and Ansgar Bernardi. 2007. Identityrank: Named entity disambiguation in the context of the NEWS project. In *Proceedings of the 4th European conference on The Semantic Web: Research and Applications*, ESWC '07, pages 640–654.

Norberto Fernández, Jesus A. Fisteus, Luis Sánchez, and Eduardo Martín. 2010. WebTLab: A cooccurrence-based approach to KBP 2010 entity-linking task. In TAC (2010).

Samuel Fernando and Mark Stevenson. 2012. Adapting wikification to cultural heritage. In *Proceedings of the 6th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 101–106.

Paolo Ferragina and Ugo Scaiella. 2010. TAGME: on-the-fly annotation of short text fragments (by Wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1625–1628.

Seeger Fisher, Aaron Dunlop, Brian Roark, Yongshun Chen, and Joshua Burmeister. 2009. OHSU summarization and entity linking systems. In TAC (2009).

Antske Fokkens, Marieke van Erp, Marten Postma, Ted Pedersen, Piek Vossen, and Nuno Freire. 2013. Offspring from reproduction problems: What replication failure teaches us. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1691–1701.

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. FACC1: Freebase annotation of clueweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0). Technical report, Google.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992a. One sense per discourse. In *Proceedings of the 4th DARPA Speech and Natural Language Workshop*, pages 233–237.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992b. Work on statistical methods for word sense disambiguation. In *Proceedings of the AAAI Fall Symposium on Intelligent Probabilistic Approaches to Natural Language*, pages 54–60.

James J. Gardner and Li Xiong. 2009. Automatic link detection: a sequence labeling approach. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1701–1704.

Matt Gardner. 2012. Adding distributional semantics to knowledge base entities through web-scale entity linking. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 46–51.

Nikesh Garera and David Yarowsky. 2009. Structural, transitive and latent models for biographic fact extraction. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 300–308.

Alan Goldschen, Brian Kapp, Tim Meyer, Boyan Onyshkevych, and Pat Schone. 2010. TCAR at TAC-KBP-2010 U.S. Department of Defense. In TAC (2010).

Jun Gong and Douglas W. Oard. 2009. Selecting hierarchical clustering cut points for web person-name disambiguation. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 778–779.

Chung Heong Gooi and James Allan. 2004. Cross-document coreference on a large scale corpus. In *HLT-NAACL 2004: Main Proceedings*, pages 9–16.

Swapna Gottipati and Jing Jiang. 2011. Linking entities to a knowledge base with query expansion. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 804–813.

David Graus, Tom Kenter, Marc Bron, Edgar Meij, and Maarten de Rijke. 2012. Context-based entity linking - University of Amsterdam at TAC 2012. In TAC (2012).

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013a. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1020–1030.

Yuhang Guo, Wanxiang Che, Ting Liu, and Sheng Li. 2011. A graph-based method for entity linking. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1010–1018.

Yuhang Guo, Bing Qin, Ting Liu, and Sheng Li. 2013b. Microblog entity linking by leveraging extra posts. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 863–868.

Ben Hachey, Will Radford, and James R. Curran. 2011. Graph-based named entity linking with Wikipedia. In *Proceedings of the 12th International Conference on Web Information System Engineering*, pages 213–226.

Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. 2013. Evaluating entity linking with Wikipedia. *Artificial Intelligence*, 194:130–150.

Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1152–1161.

Aria Haghighi and Dan Klein. 2010. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 385–393.

Hannaneh Hajishirzi, Leila Zilles, Daniel S. Weld, and Luke Zettlemoyer. 2013. Joint coreference resolution and named-entity linking with multi-pass sieves. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 289–299.

Xianpei Han and Le Sun. 2011. A generative entity-mention model for linking entities with knowledge base. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 945–954.

Xianpei Han and Le Sun. 2012. An entity-topic model for entity linking. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 105–115.

Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 765–774.

Xianpei Han and Jun Zhao. 2009a. Named entity disambiguation by leveraging Wikipedia semantic knowledge. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 215–224.

Xianpei Han and Jun Zhao. 2009b. NLPR_KBP in TAC 2009 KBP track: A two-stage method to entity linking. In TAC (2009).

Xianpei Han and Jun Zhao. 2010. Structural semantic relatedness: A knowledge-based method to named entity disambiguation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 50–59.

Joseph Hassell, Boanerges Aleman-Meza, and I. Budak Arpinar. 2006. Ontology-driven automatic entity disambiguation in unstructured text. In *Proceedings of the 5th international conference on The Semantic Web*, ISWC'06, pages 44–57.

Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013a. Learning entity representation for entity disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–34.

Zhengyan He, Shujie Liu, Yang Song, Mu Li, Ming Zhou, and Houfeng Wang. 2013b. Efficient collective entity linking with stacking. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 426–435.

Zhengyan He and Houfeng Wang. 2011. Collective entity linking and a simple slot filling method for TAC-KBP 2011. In TAC (2011).

Lynette Hirschman, Marc Colosimo, Alexander Morgan, and Alexander Yeh. 2005. Overview of BioCreAtIvE task 1B: Normalized gene lists. *BMC Bioinformatics*, 6(Supplement 1):S11.

Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. 2012. KORE: keyphrase overlap relatedness for entity disambiguation.

In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 545–554.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792.

Matthew Honnibal and Robert Dale. 2009. DAMSEL: The DSTO/Macquarie system for entity-linking. In TAC (2009).

M. Shahriar Hossain, Patrick Butler, Arnold P. Boedihardjo, and Naren Ramakrishnan. 2012. Storytelling in entity networks to support intelligence analysts. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 1375–1383.

Wei Che Huang, Shlomo Geva, and Andrew Trotman. 2010. Overview of the INEX 2009 link the wiki track. In *Proceedings of the Focused retrieval and evaluation, and 8th international conference on Initiative for the evaluation of XML retrieval*, INEX'09, pages 312–323. Springer-Verlag, Berlin, Heidelberg.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1148–1158.

Heng Ji, Ralph Grishman, and Hoa Trang Dang. 2011. Overview of the TAC 2011 knowledge base population track. In TAC (2011).

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the TAC 2010 knowledge base population track. In TAC (2010).

Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of the 12th International Conference on Knowledge Discovery and Data Mining*, pages 217–226.

Thomas Kailath. 1967. The Divergence and Bhattacharyya Distance Measures in Signal Selection. *Communication Technology, IEEE Transactions on*, 15(1):52–60.

Rianne Kaptein, Pavel Serdyukov, Arjen De Vries, and Jaap Kamps. 2010. Entity ranking using Wikipedia as a pivot. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 69–78.

H. Kucera and W. N. Francis. 1967. *Computational analysis of present-day American English*. Brown University Press, Providence, RI.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2009. Collective annotation of Wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 457–466.

Jonathan K. Kummerfeld, David Hall, James R. Curran, and Dan Klein. 2012. Parser showdown at the Wall Street corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1048–1059.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289.

John Lehmann, Sean Monahan, Luke Nezda, Arnold Jung, and Ying Shi. 2010. LCC approaches to knowledge base population at TAC 2010. In TAC (2010).

Jochen L. Leidner. 2004. Toponym resolution in text (abstract only): "which Sheffield is it?". In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 602–602.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, SIGDOC '86, pages 24–26.

Fangtao Li, Zhicheng Zheng, Fan Bu, Yang Tang, Xiaoyan Zhu, and Minlie Huang. 2009. THU QUANTA at TAC 2009 KBP and RTE track. In TAC (2009).

Xin Li, Paul Morie, and Dan Roth. 2004. Robust reading: Identification and tracing of ambiguous names. In *HLT-NAACL 2004: Main Proceedings*, pages 17–24.

Yang Li, Chi Wang, Fangqiu Han, Jiawei Han, Dan Roth, and Xifeng Yan. 2013. Mining evidences for named entity disambiguation. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '13, pages 1070–1078.

Thomas Lin, Mausam, and Oren Etzioni. 2012a. Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 84–88.

Thomas Lin, Mausam, and Oren Etzioni. 2012b. No noun phrase left behind: Detecting and typing unlinkable entities. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 893–903.

Xiao Ling, Sameer Singh, and Daniel S. Weld. 2014. Context representation for named entity linking. In *Pacific Northwest Regional NLP Workshop (NW-NLP)*.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *AAAI*, pages 94–100.

Christina Lioma, Alok Kothari, and Hinrich Schuetze. 2011. Sense discrimination for physics retrieval. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 1101–1102.

Xiaohua Liu, Yitong Li, Haocheng Wu, Ming Zhou, Furu Wei, and Yi Lu. 2013. Entity linking for tweets. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1304–1311.

Xiaoqiang Luo and Imed Zitouni. 2005. Multi-lingual coreference resolution with syntactic features. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 660–667.

Hassan H. Malik, Ian MacGillivray, Måns Olof-Ors, Siming Sun, and Shailesh Saroha. 2011. Exploring the corporate ecosystem with a semi-supervised entity graph. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 1857–1866.

Gideon Mann and David Yarowsky. 2003. Unsupervised personal name disambiguation. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 33–40.

Arturas Mazeika, Tomasz Tylenda, and Gerhard Weikum. 2011. Entity timelines: visual analytics and named entity evolution. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 2585–2588.

Andrew McCallum, Kamal Nigam, and Lyle H. Ungar. 2000. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining*, pages 169–178.

Paul McNamee. 2010. HLTCOE Efforts in Entity Linking at TAC KBP 2010. In TAC (2010).

Paul McNamee, Mark Dredze, Adam Gerber, Nikesh Garera, Tim Finin, James Mayfield, Christine Piatko, Delip Rao, David Yarowsky, and Markus Dreyer. 2009a. HLTCOE approaches to knowledge base population at TAC 2009. In TAC (2009).

Paul McNamee, Heather Simpson, and Hoa Trang Dang. 2009b. Overview of the TAC 2009 Knowledge Base Population Track. In TAC (2009).

Paul McNamee, Veselin Stoyanov, James Mayfield, Tim Finin, Tim Oates, Tan Xu, Douglas W. Oard, and Dawn Lawrie. 2012. HLTCOE participation at TAC 2012: Entity linking and cold start knowledge base construction. In TAC (2012).

Pablo N. Mendes, Joachim Daiber, Max Jakob, and Christian Bizer. 2011. Evaluating DBpedia Spotlight for the TAC-KBP entity linking task. In TAC (2011).

Charles F. Meyer. 1992. *Apposition in Contemporary English*. Cambridge University Press, Cambridge, UK.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 233–242.

George A. Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38:39–41.

David Milne and Ian H. Witten. 2008. Learning to link with Wikipedia. In *Proceedings of the 17th Conference on Information and Knowledge Management*, pages 509–518.

Einat Minkov, William W. Cohen, and Andrew Y. Ng. 2006. Contextual search and name disambiguation in email using graphs. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 27–34.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. 2010. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 788–796.

Hrushikesh Mohapatra, Siddhanth Jain, and Soumen Chakrabarti. 2013. Joint bootstrapping of corpus annotations and entity types. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 436–446.

Dan Moldovan, Christine Clark, Sanda Harabagiu, and Steve Maiorano. 2003. Cogex: A logic prover for question answering. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 87–93.

Sean Monahan and Dean Carpenter. 2012. Lorify: A knowledge base from scratch. In TAC (2012).

Sean Monahan, John Lehmann, Timothy Nyberg, Jesse Plymale, and Arnold Jung. 2011. Cross-lingual cross-document coreference with entity linking. In TAC (2011).

Alexander A. Morgan, Zhiyong Lu, Xinglong Wang, Aaron M. Cohen, Juliane Fluck, Patrick Ruch, Anna Divoli, Katrin Fundel, Robert Leaman, Jörg Hakenberg, Chengjie Sun, Heng-hui Liu, Rafael Torres, Michael Krauthammer, William W. Lau, Hongfang Liu, Chun-Nan Hsu, Martijn Schuemie, K. Bretonnel Cohen, and Lynette Hirschman. 2008. Overview of BioCreative II gene normalization. *Genome Biology*, 9(Supplement 2):S3.

Ndapandula Nakashole, Tomasz Tylenda, and Gerhard Weikum. 2013. Fine-grained semantic typing of emerging entities. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1488–1497.

Vivi Nastase, Alex Judea, Katja Markert, and Michael Strube. 2012. Local and global context for supervised and unsupervised metonymy resolution. In *Proceedings*

of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 183–193.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41:10:1–10:69.

Ani Nenkova, Advaith Siddharthan, and Kathleen McKeown. 2005. Automatically learning cognitive status for multi-document summarization of newswire. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 241–248.

Cheng Niu, Wei Li, and Rohini K. Srihari. 2004. Weakly supervised learning for cross-document person name disambiguation supported by information extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 597–604.

Joel Nothman. 2014. *Grounding event references in news*. Ph.D. thesis, School of Information Technologies, University of Sydney.

Joel Nothman, Matthew Honnibal, Ben Hachey, and James R. Curran. 2012. Event linking: Grounding event reference in a news archive. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–232.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R. Curran. 2013. Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence*, 194:151–175.

Ted Pedersen. 2008. Empiricism is not a matter of faith. *Computational Linguistics*, 34(3):465–470.

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 Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Slav Petrov and Dan Klein. 2007. Learning and inference for hierarchically split PCFGs. In *Proceedings of the 22nd AAAI Conference of Artificial Intelligence*, pages 1642–1645.

Anja Pilz and Gerhard Paaß. 2012. Collective search for concept disambiguation. In *Proceedings of COLING 2012*, pages 2243–2258.

Glen Pink, Will Radford, Will Cannings, Andrew Naoum, Joel Nothman, Daniel Tse, and James R. Curran. 2013. SYDNEY_CMCRC at TAC 2013. In TAC (2013).

Danuta Ploch. 2011. Exploring entity relations for named entity disambiguation. In *Proceedings of the ACL 2011 Student Session*, pages 18–23.

Danuta Ploch, Leonhard Hennig, Ernesto William De Luca, and Sahin Albayrak. 2011. DAI approaches to the TAC-KBP 2011 entity linking task. In TAC (2011).

Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with Markov Logic. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 650–659.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. General Grammar Series. Longman, London, UK.

Will Radford, Will Cannings, Andrew Naoum, Joel Nothman, Glen Pink, Daniel Tse, and James R. Curran. 2012. (Almost) Total Recall – SYDNEY_CMCRC at TAC 2012. In TAC (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 (Volume 2: Short Papers)*, pages 671–677.

Will Radford, Ben Hachey, Matthew Honnibal, Joel Nothman, and James R. Curran. 2011. Naive but effective NIL clustering baselines – CMCRC at TAC 2011. In TAC (2011).

Will Radford, Ben Hachey, Joel Nothman, Matthew Honnibal, and James R. Curran. 2010. CMCRC at TAC10: Document-level entity linking with graph-based reranking. In TAC (2010).

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nate 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.

Altaf Rahman and Vincent Ng. 2011. Coreference resolution with world knowledge. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 814–824.

Delip Rao, Nikesh Garera, and David Yarowsky. 2007. JHU1: An unsupervised approach to person name disambiguation using web snippets. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 199–202.

Delip Rao, Paul McNamee, and Mark Dredze. 2010. Streaming cross document entity coreference resolution. In *Coling 2010: Posters*, pages 1050–1058.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155.

Lev Ratinov and Dan Roth. 2011. GLOW TAC-KBP2011 entity linking system. In TAC (2011).

Lev Ratinov and Dan Roth. 2012. Learning-based multi-sieve co-reference resolution with knowledge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1234–1244.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1375–1384.

Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. 2013. The life and death of discourse entities: Identifying singleton mentions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 627–633.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420.

Dan Roth and Mark Sammons. 2007. Semantic and logical inference model for textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 107–112.

Robert E. Schapire and Yoram Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168.

Barry Schiffman, Inderjeet Mani, and Kristian Concepcion. 2001. Producing biographical summaries: Combining linguistic knowledge with corpus statistics. In

Proceedings of 39th Annual Meeting of the Association for Computational Linguistics, pages 458–465.

Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2012a. A graph-based approach for ontology population with named entities. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pages 345–354.

Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2012b. LINDEN: Linking named entities with knowledge base via semantic knowledge. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 449–458.

Advaith Siddharthan. 2002. Resolving attachment and clause boundary ambiguities for simplifying relative clause constructs. In *Proceedings of the ACL Student Research Workshop (ACLSRW 2002)*, pages 60–65.

Avirup Sil, Ernest Cronin, Penghai Nie, Yinfei Yang, Ana-Maria Popescu, and Alexander Yates. 2012. Linking named entities to any database. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 116–127.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2011. Large-scale cross-document coreference using distributed inference and hierarchical models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 793–803.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical Report UM-CS-2012-015, University of Massachusetts, Amherst.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Valentin I. Spitkovsky and Angel X. Chang. 2012. A cross-lingual dictionary for english Wikipedia concepts. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 3168–3175.

Vivek Srikumar, Roi Reichart, Mark Sammons, Ari Rappoport, and Dan Roth. 2008. Extraction of entailed semantic relations through syntax-based comma resolution. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1030–1038.

Rosa Stern, Benoît Sagot, and Frédéric Béchet. 2012. A joint named entity recognition and entity linking system. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, pages 52–60.

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.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2003. An improved extraction pattern representation model for automatic ie pattern acquisition. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 224–231.

Suzanne Tamang, Zheng Chen, and Heng Ji. 2012. CUNY-BLENDER TAC-KBP2012 Entity Linking System and Slot Filling Validation System. In TAC (2012).

Bilyana Taneva, Mouna Kacimi, and Gerhard Weikum. 2011. Finding images of difficult entities in the long tail. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 189–194.

NIST. 2005. The ACE 2005 (ACE05) evaluation plan.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the 7th Conference on Natural Language Learning at HTL-NAACL 2003*, pages 142–147.

Olga Uryupina, Massimo Poesio, Claudio Giuliano, and Kateryna Tymoshenko. 2011. Disambiguation and filtering methods in using web knowledge for coreference resolution. In *FLAIRS Conference*, pages 317–322.

David Vadas and James R. Curran. 2007. Parsing internal noun phrase structure with collins' models. In *Proceedings of the Australasian Language Technology Workshop 2007*, pages 109–116.

Vasudeva Varma, Praveen Bysani, Kranthi Reddy, Vijay Bharath Barath, and Sudheer Kovelamudi. 2010. IIIT Hyderabad in Guided Summarization and Knowledge Base Population. In TAC (2010).

Vasudeva Varma, Praveen Bysani, Kranthi Reddy, Vijay Bharat, Santosh GSK, Karuna Kumar, Sudheer Kovelamudi, Kiran Kumar N, and Nitin Maganti. 2009. IIIT Hyderabad at TAC 2009. In TAC (2009).

Nina Wacholder, Yael Ravin, and Misook Choi. 1997. Disambiguation of proper names in text. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 202–208.

Daisy Zhe Wang, Yang Chen, Sean Goldberg, Christan Grant, and Kun Li. 2012. Automatic knowledge base construction using probabilistic extraction, deductive reasoning, and human feedback. In *Proceedings of the Joint Workshop on Automatic Knowl-*

edge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX), pages 106–110.

Yankai Wang, Yan Lin, Zhiyuan Liu, and Maosong Sun. 2013. THUNLP at TAC KBP 2013 in entity linking. In TAC (2013).

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2011. OntoNotes Release 4.0. Technical report, Linguistic Data Consortium, Philadelphia, PA USA.

Michael White and Rajakrishnan Rajkumar. 2008. A more precise analysis of punctuation for broad-coverage surface realization with CCG. In *Coling 2008: Proceedings of the workshop on Grammar Engineering Across Frameworks*, pages 17–24.

Michael Wick, Karl Schultz, and Andrew McCallum. 2012. Human-machine cooperation: Supporting user corrections to automatically constructed kbs. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 89–94.

William E. Winkler. 2006. Overview of record linkage and current research directions. Technical report, Bureau of the Census.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2013. Universal schema for entity type prediction. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, AKBC '13, pages 79–84.

Jingtao Yu, Omkar Mujgond, and Rob Gaizauskas. 2010. The University of Sheffield system at TAC KBP 2010. In TAC (2010).

Hugo Zaragoza, Henning Rode, Peter Mika, Jordi Atserias, Massimiliano Ciaramita, and Giuseppe Attardi. 2007. Ranking very many typed entities on Wikipedia. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 1015–1018.

Wei Zhang, Chuan Sim Sim, Jian Su, and Chew-Lim Tan. 2011a. Entity linking with effective acronym expansion, instance selection and topic modelling. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 1909–1914.

Wei Zhang, Jian Su, Bin Chen, Wenting Wang, Zhiqiang Toh, Yanchuan Sim, Yunbo Cao, Chin Yew Lin, and Chew Lim Tan. 2011b. I2R-NUS-MSRA at TAC 2011: Entity Linking. In TAC (2011).

Wei Zhang, Jian Su, and Chew-Lim Tan. 2011c. A Wikipedia-LDA model for entity linking with batch size changing instance selection. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 562–570.

Wei Zhang, Jian Su, Chew-Lim Tan, Yunbo Cao, and Chin-Yew Lin. 2012. A lazy learning model for entity linking using query-specific information. In *Proceedings of COLING 2012*, pages 3089–3104.

Wei Zhang, Jian Su, Chew Lim Tan, and Wen Ting Wang. 2010. Entity linking leveraging automatically generated annotation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1290–1298.

Yu Zhao, Weipeng He, Zhiyuan Liu, and Maosong Sun. 2011. THUNLP at TAC KBP 2011 in entity linking. In TAC (2011).

Jiaping Zheng, Luke Vilnis, Sameer Singh, Jinho D. Choi, and Andrew McCallum. 2013. Dynamic knowledge-base alignment for coreference resolution. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 153–162.

Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. 2010. Learning to link entities with knowledge base. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 483–491.

Zhicheng Zheng, Xiance Si, Fangtao Li, Edward Y. Chang, and Xiaoyan Zhu. 2012. Entity disambiguation with freebase. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '12, pages 82–89.

Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. 2013. Do we need entity-centric knowledge bases for entity disambiguation? In *Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies*, i-Know '13, pages 4:1–4:8.