

Web Knowledge Bases

Andrew Chisholm

Supervisor: Ben Hachey



A thesis submitted
in fulfilment of the requirements
for the degree of Doctor of Philosophy
in the School of Information Technologies at
The University of Sydney
School of Information Technologies
2018

Abstract

Knowledge is key to natural language understanding. References to specific people, places and things in text are crucial to resolving ambiguity and extracting meaning. Knowledge Bases (KBs) codify this information for automated systems — enabling applications such as entity-based search and question answering. This thesis explores the idea that sites on the web may *act* as a KB, even if that is not their primary intent.

Dedicated KBs like Wikipedia are a rich source of entity information, but are built and maintained at an ongoing cost in human effort. As a result, they are generally limited in terms of the breadth and depth of knowledge they index about entities. Web knowledge bases offer a distributed solution to the problem of aggregating entity knowledge. Social networks aggregate content about people, news sites describe events with tags for organizations and locations, and a diverse assortment of web directories aggregate statistics and summaries for long-tail entities notable within niche movie, musical and sporting domains. We aim to develop the potential of these resources for both web-centric entity Information Extraction (IE) and structured KB population.

We first investigate the problem of Named Entity Linking (NEL), where systems must resolve ambiguous mentions of entities in text to their corresponding node in a structured KB. We demonstrate that entity disambiguation models derived from inbound web links to Wikipedia are able to complement and in some cases completely replace the role of resources typically derived from the KB. Building on this work, we observe that any page on the web which reliably disambiguates inbound web links may act as an aggregation point for entity knowledge. To uncover these resources, we formalize the task of Web Knowledge Base Discovery (KBD) and develop a system to automatically infer the existence of KB-like endpoints on the web. While extending our framework to multiple KBs increases the breadth of available entity knowledge, we must still consolidate references to the *same* entity across *different* web KBs. We investigate this task of Cross-KB Coreference Resolution (KB-Coref) and develop models for efficiently clustering coreferent endpoints across web-scale document collections.

Finally, assessing the gap between unstructured web knowledge resources and those of a typical KB, we develop a neural machine translation approach which transforms entity knowledge between unstructured textual mentions and traditional KB structures.

The web has great potential as a source of entity knowledge. In this thesis we aim to first discover, distill and finally transform this knowledge into forms which will ultimately be useful in downstream language understanding tasks.

Acknowledgements

First I'd like to thank my supervisors. Foremost thanks go to Ben Hachey (the OG) who has worn many hats throughout the process, always in style. Your good grace, wisdom, and humor in all things have made working toward this thesis easy, especially when the work itself was hard. Thankyou to Will Radford, who has been a brilliant supervisor, colleague and friend. I have learnt much from your thoughtful reviews, advice and pragmatism both at work and in research. If Skynet does take over, I know I'll have received fair warning via arxiv.org links punctuated with emoji. Thanks also go to James Curran, who first welcomed me into ə-lab and made it feel like home. And finally Alan Fekete, who gracefully bore the brunt of bureaucracy when the need arose, you have my apologies and gratitude in equal measure.

I've always preferred to write code instead of text and must sincerely thank those who provided an excuse. For the early days, I must thank John Dunbar and Dave Bevin for making my initial escape into academia possible. Thanks also go to Whitney Komor and Darren Chait who both saw a path between academic and real-world applications, and my colleagues Anaïs Cadilhac, Bo Han, Steve Strickland and Louis Rankin who each helped shape the path my work took at different points.

Thankyou to my colleges and friends amongst the ə-lab diaspora. In particular to Joel Nothman for his advice and thoughtful reviews and Tim Dawborn for always being available to help put out technical fires¹. Thanks also go to Glen Pink, Kellie

¹Special credits to schwa11 — I hope you too can read this one day

Webster, Dominick Ng, Andrew Naoum and Nicky Ringland who each helped through reading groups, review parties, lunches and coffee runs.

For my friends and family, special thanks go to all those who didn't ask when I'd finish this thing, and apologies to all those who believed one of the answers. Thankyou to my parents Paul and Janine, your support made this work possible. And finally, thank you to Jess, for both believing in and joining me on this adventure.

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: *Andrew Chisholm*

Signature:

Date: *8th October 2018*

Contents

1	Introduction	3
1.1	Web Knowledge Bases	5
1.2	Cross-KB Coreference	7
1.3	Transforming entity knowledge	8
1.4	Generating entity descriptions	9
1.5	Contributions and Outline	11
2	Background	13
2.1	Recognition of named entities in text	14
2.2	Disambiguating entity mentions	15
2.2.1	Coreference clustering	16
2.2.2	Entity linking	19
2.3	Stores of entity knowledge	20
2.4	Extracting structured knowledge	23
2.5	Learned representations	26
2.5.1	Text representation	28
2.5.2	Representing entities and relations	29
2.6	Summary	31
3	Entity Disambiguation with Web Links	33
3.1	Introduction	34
3.2	Related work	36

3.3	Terminology	38
3.4	Tasks	39
3.4.1	CoNLL	39
3.4.2	TAC 2010	41
3.5	Wikipedia benchmark models	42
3.5.1	Entity prior	42
3.5.2	Name probability	43
3.5.3	Textual context	44
3.6	Web link models	46
3.6.1	Entity prior	46
3.6.2	Name probability	47
3.6.3	Textual context	48
3.7	Learning to rank	49
3.7.1	Feature selection	50
3.7.2	Effect of training data size	52
3.7.3	Ablation analysis	52
3.8	Adding coherence	53
3.8.1	A two-stage classifier	54
3.9	Final experiments	54
3.9.1	Results	55
3.10	Discussion	57
3.11	Summary	59
4	Web Knowledge Base Discovery	61
4.1	Introduction	62
4.2	Related work	65
4.3	Web entity endpoints	66
4.3.1	Online encyclopedia	66
4.3.2	Web news	66

4.3.3	Social networks	67
4.3.4	Organisation directories	67
4.4	Framework	67
4.4.1	Endpoint inference	68
4.4.2	Features	69
4.5	Dataset	71
4.6	Model	72
4.6.1	Development experiments	72
4.6.2	Analysis	73
4.7	Evaluation	75
4.7.1	Crowd task	75
4.7.2	Results	76
4.8	Discussion	77
4.9	Summary	79
5	Cross-KB Coreference Resolution	81
5.1	Introduction	82
5.2	Related work	84
5.3	Methodology	86
5.4	Datasets	87
5.4.1	Preprocessing	87
5.4.2	Statistics	91
5.5	Identifying KBs	92
5.5.1	Results	93
5.5.2	Analysis	94
5.6	Resolving link coreference	95
5.6.1	Entity representation	96
5.6.2	Features	97
5.6.3	Instance sampling	99

5.6.4	Training the model	99
5.7	Clustering	101
5.7.1	Constraints	101
5.7.2	Iterative URL aggregation	103
5.8	Evaluation	105
5.8.1	Endpoint results	106
5.8.2	Clustering results	107
5.9	Analysis	108
5.10	Discussion	109
5.10.1	Future work	110
5.11	Summary	111
6	Biography generation	113
6.1	Introduction	114
6.2	Related work	115
6.3	Task and data	118
6.3.1	Task complexity	120
6.4	Model	121
6.4.1	Sequence-to-sequence model (S2S)	122
6.4.2	S2S with autoencoding (S2S+AE)	123
6.5	Experimental methodology	124
6.5.1	Benchmarks	125
6.5.2	Metrics	125
6.5.3	Analysis of content selection	126
6.6	Results	127
6.6.1	Comparison against Wikipedia reference	127
6.6.2	Human preference evaluation	127
6.7	Analysis	128
6.7.1	Fact Count	128

6.7.2	Example generated text	129
6.7.3	Content selection and hallucination	131
6.8	Discussion	133
6.9	Summary	134
7	Fact inference	137
7.1	Introduction	138
7.2	Related work	140
7.3	Data	143
7.3.1	Facts	144
7.3.2	Text	145
7.4	Model	147
7.4.1	Sequence-to-sequence model	149
7.4.2	Preprocessing	150
7.4.3	Training	150
7.4.4	Inference	152
7.5	Results	153
7.5.1	Comparison with the Wikidata reference	153
7.5.2	Thresholding decode scores	157
7.6	Analysis	160
7.6.1	Performance vs Inlink Count	160
7.6.2	Example generated facts	161
7.6.3	Fact Explicitness	164
7.7	Discussion	165
7.8	Summary	167
8	Conclusion	169
8.1	Future Work	171
8.2	Summary	173

Bibliography

175

List of Figures

3.1	Inter-article links across pages from Wikipedia.	35
3.2	Inlinks to Wikipedia from external sources.	35
3.3	Combined Web and Wikipedia inlinks.	36
3.4	Performance of individual and cumulative combinations of model features	51
3.5	SVM learning curves for best configurations.	52
3.6	Ablation analysis of best configurations.	53
4.1	Annotated links to Tesla Motors and Elon Musk on nytimes.com	63
4.3	Links targeting entity endpoints across multiple web KBs.	64
4.4	Precision-recall trade-off across thresholds.	73
4.5	CrowdFlower annotation interface for endpoint URL evaluation	77
5.1	Three web pages representing the same Tesla Motors entity	82
5.2	Two web pages representing distinct Mikael Petersen entities.	83
5.3	Agglomeration of endpoint clusters via pairwise mention set comparisons	101
5.4	Frequency vs. Rank for link targets referencing “Obama” across Wikipedia	104
6.1	Example of linearized Wikidata fact encoding	115
6.2	Sequence-to-sequence translation from linearized facts to text.	122
6.3	Sequence-to-sequence autoencoder.	124
6.4	BLEU vs Fact Count on instances from DEV	129
7.1	Multi-fact inference over a shared input representation	139

7.2	Multi-output sequence to sequence relation inference model	148
7.3	Macro-averaged Precision vs Recall for each model.	158
7.4	Precision vs Recall across fact-types	158
7.5	Performance vs Inlink Count	160

List of Tables

1.1	Sample of Wikidata facts for Elon Musk	10
3.1	Summary of evaluation datasets for entity disambiguation	40
3.2	Performance of individual feature components derived from each source	43
3.3	Comparison of page-entity link graphs	47
3.4	Comparison of name-entity link graphs	48
3.5	Coverage of textual context models over entities by source	49
3.6	In-vocab tokens per entity for each context model	49
3.7	Web link components vs. Wikipedia.	55
3.8	Web link combinations vs. Wikipedia.	56
3.9	Web links complement Wikipedia.	56
3.10	Comparison to the disambiguation literature.	57
4.1	Example of path features generated for sample URLs	70
4.2	Top mention-aligned URL prefixes in the seed corpus.	71
4.3	Summary statistics for the HG-NEWS corpus	72
4.4	Model estimates for notable endpoints	74
4.5	Sample of predicted URL patterns and entity counts.	74
5.1	Summary statistics for each dataset.	91
5.2	Comparison of links to English Wikipedia across web corpora.	92
5.3	Model performance on the mention prediction task	93
5.4	Statistics of high-confidence entity links extracted from each dataset. . .	94

5.5	Sample of top-10 endpoint patterns by unique inlink count.	94
5.6	Sample mentions and extracted features for coreference classification .	98
5.7	Statistics of the sampled coreference training corpus	100
5.8	Number of input URLs and output clusters for each corpus.	105
5.9	Distribution of entity-link types extracted by KBD over annotated samples.	106
5.10	Coreference clustering results	107
5.11	Discovered coreference clusters for Tesla Motors and Nikola Tesla	108
6.1	Top populated facts for human entities in Wikidata	119
6.2	Language model perplexity across templated datasets.	121
6.3	BLEU scores for each hypothesis against the Wikipedia reference	127
6.4	Human preference evaluation results	128
6.5	Sample of input facts and corresponding output text for each model . .	130
6.6	Content selection evaluation results	132
6.7	Fact density and sentence length analysis.	132
6.8	Analysis of hallucinated facts in text generated by model	133
7.1	Summary of relations, output vocabulary size and baseline performance	145
7.2	Sample of mentions from inlinks to the Elon Musk Wikipedia article . .	146
7.3	Relations populated for the Elon Musk entity in Wikidata.	147
7.5	Precision of the LNK fact inference model trained and evaluated on sentences linking to the entities from the TEST set.	156
7.6	Sample of mentions across confidence thresholds.	159
7.7	Sample of input mentions and corresponding output facts	163

1 Introduction

The web is a vast and rapidly growing store of information. For systems seeking to harness human knowledge, natural language on the web represents a valuable resource. In contrast to knowledge stores like Freebase or Wikipedia, content on the web is predominately unstructured. For this content to be useful in automated systems, we must first distill contained knowledge from its latent form in text. If we can bridge this gap, web sourced knowledge presents an appealing alternative to manual knowledge curation. Content from the web is typically generated as by-product of existing commercial and user driven web publishing activity. Moreover, domain coverage is as broad, deep and up-to-date as the interests of its users.

For applications which rely on these resources, coverage and currency is often critically important. Question Answering (QA) in particular has become an important feature of virtual assistant products like Apple Siri and Google Now. Where these products rely on human-curated knowledge, they cannot rapidly adapt to changes in the real world. For example, we may reasonably pose a question along the lines of "How fast is the new Tesla Roadster?" on the same day it is announced. While this information is readily available in the form of unstructured text in press releases and news coverage across the web, dedicated knowledge stores are typically far slower to review and incorporate specific facts of this form. This problem is exacerbated for entities and facts at the tail end of the notability distribution where the effort of dedicated KB curators is rarely spent.

Given we can in principle reproduce much of the content of structured KBs from information on the web, systems which address this challenge have long been a focus of work in Information Extraction (IE). For systems in this domain, the links *between* pages on the web often encode valuable semantic cues. Given their central role in the definition of the web itself, it's no surprise that links have long been studied both for the graphical structure they imbue upon the web (Broder et al., 2000) and what they can imply about the relation and relative importance of linked resources (Page et al., 1998). For pages representing Named Entities – i.e. people, places and things from the real world, links present a direct opportunity to extract knowledge.

When references to entities on the web coincide with outgoing web links, we may leverage both the content and textual context of these links to help resolve some of the hardest problems facing generalized information extraction systems. For example:

(1) Today [Tesla](en.wikipedia.org/wiki/Tesla_Motors) announced . . .

The presence of a web link in Example 1 helps resolve two fundamental problems in language understanding. First, the anchor span “Tesla” delineates the bounds of a named entity mention in text. Where the page targeted by a link represents an entity, anchors mark references to that entity in text. This convention for navigation on the web therefore also provides a weak form of annotation for the otherwise challenging task of Named Entity Recognition (NER).

In addition we may leverage the specific link target `Tesla_Motors` as a source of knowledge. While NER is able to delineate the name of an entity, systems still face a challenge in resolving name ambiguity. In the case of “Tesla”, the name is shared by both the electric vehicle company Tesla Motors and the inventor Nikola Tesla for which it is named. While names are generally ambiguous, links may act as kind of unique entity identifier when present, thereby resolving the task of Named Entity Disambiguation (NED).

While this style of linking is common on the web, the example above represents an ideal case. In practice there is still great variety in how links are used. Not all link

anchors constitute named entity mentions and not all entity mentions are wrapped by links. Similarly, link targets do not always resolve ambiguity or even necessarily address the same entity being referenced. Despite these constraints, it's clear that if even a small fraction of linked text on the web yields valuable semantic information, we would be left with a huge resource for knowledge-intensive tasks.

1.1 Web Knowledge Bases

Wikipedia and other dedicated KBs are a natural aggregation point for links referencing entities on the web. They are however not unique in their function as a disambiguation endpoint for inbound links. Consider a variation on the example above:

(2) Today [Tesla](nytimes.com/topic/company/tesla-motors-inc) announced . . .

In this case, the URL target of the link references a page designed to aggregate news articles about an entity. Regardless, this link is sufficient to uniquely identify the entity being referenced and thus provide a valuable semantic cue. We refer to URLs which exhibit this disambiguation property as **entity endpoints**. This design pattern is common on the web and presents an opportunity for extending link-driven information extraction beyond the bounds of traditional KBs. By relaxing the definition of a KB to any endpoint which reliably disambiguates inbound web links, we may leverage a huge variety of KB-like structure on the web.

The structure implied by inlinks to these endpoints often resembles that of a traditional KB, though the effort spent in annotating these mentions is generally motivated by standard web publishing concerns, i.e. driving traffic between news stories and optimizing a site for search engine discoverability. These resources simultaneously present solutions to the hard problems of entity coverage and update latency that face traditional monolithic NED systems. By integrating inlinks to wide-domain KBs like Wikipedia with resources that focus on deep coverage of a narrow domains like

IMDb¹ or MusicBrainz², we can combine the coverage of discrete KBs. We also may address less-notable entities which do not otherwise meet the notability constraints of a standard KB, e.g. those covered in organisation directories for small companies or via social media profiles. In leveraging resources which are both distributed and constantly updated (e.g. news and social sites), we reduce the latency at which previously unseen entities are discovered and integrated into a live system. For example, consider a continuation of the snippet above:

(3) ...announced details for its new [Roadster](tesla.com/roadster) at a press event ...

Given knowledge that URLs of the form `tesla.com/*` represent entities — in this case, products of the Tesla Motors company — we may infer the existence of a new and emerging entity from occurrence of the previously unseen identifier: `roadster` .

We refer to the task of identifying URLs which reliably disambiguate entity mentions as **Knowledge Base Discovery** (KBD). Formally, KBD takes a set of documents annotated with web links and returns a set of URL patterns which specify entity endpoints. For the examples described above, we would expect to retrieve patterns such as:

```
en.wikipedia.org/wiki/*
nytimes.com/topic/company/*
tesla.com/*
```

Identifying endpoint patterns is a non-trivial task. In some cases, the structure of a URL path itself can yield clues (e.g. the presence of `/wiki/` or `/person/` in the path), but for many endpoints, some prior knowledge that a pattern references entities is needed (e.g. `twitter.com/*`). Moreover, many patterns which resemble entity endpoints do not reference named entities and must be filtered e.g. `nytimes.com/yyyy/mm/dd/business/*` .

Given a mechanism for identifying these resources, we may potentially address the entire web as a target for entity resolution. These links and content they annotate may then be applied to downstream information extraction tasks.

¹<http://www.imdb.com>

²<https://musicbrainz.org>

1.2 Cross-KB Coreference

While we can certainly improve the breadth of entity coverage through the discovery of diverse web KBs, we cannot improve depth without a mechanism for consolidating coreferent entity references. Consider the following URLs, all of which cover the same underlying entity:

```
twitter.com/teslamotors  
en.wikipedia.org/wiki/Tesla_Motors  
nytimes.com/topic/company/tesla-motors-inc
```

While KBD may identify these endpoints, we have no way to automatically reconcile and consolidate coreferent records across KBs. KBD is thus necessary but not sufficient for end-to-end web KB construction. We refer to the task of clustering endpoint URLs which reference the same underlying entity as **Cross-KB Coreference Resolution (KB-Coref)**. KB-Coref systems may leverage information present in an entity URL, the content of the page targeted by the URL, or the content around inlinks for a URL on the web in resolving coreference between entity records. In this final case, the problem of KB-Coref closely resembles that of standard NED, where instead of resolving a single mention, we seek to resolve a cluster of coreferent mentions against the KB.

KBD and KB-Coref together form the basic building blocks for web KB construction. Given a corpus of documents from the web, we can extract a set of endpoint URLs via KBD. For every URL discovered we potentially recover thousands of inlinks representing mentions of that entity on the web. To consolidate entity reference across KBs, we next cluster together URLs which reference the same entity across distinct KB endpoints through KB-Coref. As new documents are introduced into the corpus over time, new entities are observed as instances matching existing endpoint patterns and new information about existing entities is aggregated via inlinks to existing URL clusters. This approach to KB construction is a light-weight alternative to the structured, top-down design of traditional KBs and has the potential to both cover a wider

domain of entities and aggregate a far greater corpus of entity knowledge from natural mentions on the web.

1.3 Transforming entity knowledge

Linked textual references are directly applicable to problems of entity reference, i.e. the recognition and disambiguation of named entity mentions. For applications of these resources to Question Answering (QA) and structured search, a canonicalized representation of entity knowledge is often required. Where traditional KBs like Wikidata and Freebase maintain a curated schema of facts for each entity, we may alternatively recover a structured knowledge from mentions of an entity in text. For example:

- (4) [Telsa](telegraph.co.uk/tesla) Chief Executive [Elon Musk](twitter.com/elonmusk) claims the new [Roadster](tesla.com/roadster) is capable of reaching 100kph in just 1.9 seconds — making it the fastest production car in the world.

This small snippet encodes a variety of useful facts about Elon Musk, Tesla Motors and the emerging Tesla Roadster entity. Written as relational triples of (entity, relation, value), we can directly observe expressed relations such as (Elon Musk, CEO, Tesla). In addition, we may *infer* implied facts such as (Roadster, instance-of, automobile) or (Tesla, produces, sports-cars). While these facts are not explicitly mentioned in text, we may nonetheless assert them with high-confidence given the information available. Structured facts provide a direct mechanism for answering questions about a given entity. For a query such as "how fast is the new Roadster?", we need only search for facts of the form (Roadster, speed, *) amongst those recovered from text.

While KBD and KB-Coref together provide a mechanism for extracting disambiguated entity mentions from the web, they do not address the problem of producing a more structured knowledge representation. This task closely resembles that of Slot Filling (SF) — a query driven version of the Relation Extraction (RE) task where systems seek to fill slots for an entity with values extracted from text. We may however view this

task as a kind of translation problem between equivalent structured and unstructured knowledge representations of an entity. In contrast to a regular Machine Translation (MT) where information is transformed between two unstructured natural language forms, we instead seek to decode structured facts from a collection of unstructured entity mentions.

Approaching the task of fact inference through the lens of translation has many benefits. While extractive systems are constrained to emit values matching textual spans from the input, a translation driven model instead *generates* fact values which are merely conditioned on the input text. This allows the model to both learn the target KB schema and generate fact values which may never be explicitly realized in text.

1.4 Generating entity descriptions

Given a mechanism for performing text-to-fact translation, it is natural to consider the inverse of this problem — that of generating text from a structured representation of entity knowledge. For encyclopedic KBs like Wikipedia, natural language is still the dominant human interface. Editors invest great effort in the maintenance of concise, informative textual summaries for entities. For example, consider the first sentence describing Elon Musk on Wikipedia:

- (5) Elon Reeve Musk (born June 28, 1971) is a South African-born Canadian-American business magnate, investor, engineer and inventor.

This short biographic summary is a dense but fluent representation of the most salient facts for the entity. The corresponding subset of Wikidata facts shown in Table 1.1 is much harder to interpret quickly.

While these two representations of entity knowledge are roughly equivalent, specific instances of information disparity may be problematic for translation. For example, we cannot hope to reproduce a reference to dual "Canadian-America" citizenship for Elon Musk without having both corresponding `citizenship` facts populated in the source

instance of	Human	given name	Elon
gender	Male	family name	Musk
date of birth	1971 06 28	citizenship	United States
occupation	Entrepreneur	birth place	South Africa

Table 1.1: Sample of Wikidata facts for Elon Musk

KB. Moreover, evaluation of generated text in terms the factual content selected and reproduced presents a challenge — one for which standard similarity based translation metrics are ill equipped.

Provided we can address these challenges, automating the process of entity summary generation has clear applications for both traditional KB curation and web-KB construction. For traditional KBs, generated summaries may be used to populate descriptions of entities where some facts are known, but no article has yet been created — for example, populating or updating articles for languages with low curator coverage from a common set of facts. For web KBs, a mechanism for summarizing indexed entities is clearly desirable for interaction with human consumers. A user may directly ask: "Tell me about the new Roadster". Given a set of facts describing the entity — sourced from an existing KB or inferred directly from mentions in text — we may simply translate the current structured representation into an ad-hoc summary of available entity knowledge. As facts are added and change over time, our translation model may be invoked again to produce up-to-date entity descriptions.

1.5 Contributions and Outline

This thesis investigates the extraction and application of large scale web-based knowledge stores to KB creation and population tasks.

Our core contributions include original system implementations for Named Entity Disambiguation (NED), Web KB Discovery (KBD), Cross-KB Coreference Resolution (KB-Coref) and neural translation models for entity text generation and fact inference. We provide a detailed analysis of system performance in comparison to existing benchmarks for NED and introduce new annotated datasets for evaluating KBD and KB-Coref. We evaluate fact-driven biography generation in terms of both content selected and human preference, and analyze precision with respect to a Wikidata reference in fact inference experiments. We also make available code and data artifacts developed as part of this thesis — including over 1.5 billion web documents with extracted text, named entities and entity endpoint URL annotations. Details of key contributions by chapter follow:

In Chapter 2, we give a broad background to information extraction problems with a focus on tasks in knowledge base population and entity-document representation. We highlight systems leveraging world knowledge resources to better address each task and summarize existing approaches to extracting and developing web knowledge resources.

In Chapter 3, we describe work on entity disambiguation with web links. We develop a NED system for Wikipedia entities using inbound web links as a knowledge source for disambiguation models. Our analysis suggest that web links can augment or even completely replace curated knowledge resources on this task. Work described in this chapter was published as a journal article in Chisholm and Hachey (2015).

In Chapter 4, we formalize and explore the task of KBD. We first develop a system which learns to infer the existence of KB endpoints from a corpus of unlabelled web documents. We then build a crowd-sourced corpus of entity endpoint annotations and

use these for evaluation. Work described in this chapter was published in workshop proceedings (Chisholm et al., 2016b).

In Chapter 5 we investigate the task of Cross-KB coreference. We build two web-scale document corpora from open-access and up-to-date web crawls and attach automated NER and KBD endpoint annotations. We then develop and evaluate a baseline entity endpoint coreference clustering system using information from inbound links on this data. Pairwise KB-Coref was the focus of the 2016 ALTA shared task (Chisholm et al., 2016a).

In Chapter 6, we explore biography generation under a neural network sequence-to-sequence translation framework. Our model transforms information from structured facts into single-sentence natural language summaries for person entities in Wikipedia. We provide a detailed analysis of fact-driven text generation, evaluating content selection and human preference alongside standard translation metrics. Biography generation work was published in conference proceedings (Chisholm et al., 2017).

In Chapter 7, we invert this translation task — transforming unstructured textual description into structured facts about an entity. We evaluate two distinct configurations, one mirroring generation experiments where facts are extracted from entity summaries, and one simulating the web KB setting with facts extracted from a dispersed sample of inbound links to an entity page.

In Chapter 8, we conclude upon work described in this thesis. We consider how a pipeline of web KB discovery, coreference resolution and knowledge translation lay the groundwork for broader integration of web resources into traditional KB population and entity knowledge tasks.

2 Background

All models are wrong, but some are useful.

George Box

A vast amount of useful knowledge is encoded as unstructured natural language. Information extraction (IE) systems seek to decode this knowledge into a form which can be used in downstream knowledge tasks. In Chapter 1, we described how links between documents on the web can provide valuable semantic knowledge to IE systems. We described how link annotations help resolve key steps in the traditional IE pipeline of entity recognition and disambiguation, and how deep learning might be applied to bridge the gap between structured and unstructured knowledge representations for entities. This approach was motivated by the potential of web links to aggregate information across a wider range of sources and cover a larger number of entities than traditional monolithic KBs.

Structured KBs and web resources have played a crucial role as knowledge sources and evaluation sets for various tasks in the IE pipeline. We highlight the role these resources have played in the development of IE systems and explore the core shared tasks and datasets which have helped formalize this work. This chapter broadly describes background work on IE, its component tasks and applications of neural networks in this domain. We will revisit related work specific to each task explored in the chapters which follow as they are introduced therein.

2.1 Recognition of named entities in text

Entity references in text ground textual meaning to objects and concepts from the real world. The problem of identifying and classifying entity references is therefore fundamental to language understanding and has been a long studied problem in NLP. Named entity recognition (NER) seeks to identify spans of text which reference entities by name (i.e. proper nouns), ignoring nominal and pronominal references. We refer to the fragments or phrases which delineate entity references as **mentions**. For example, the following sentence contains two named entity mentions:

- (6) Today [Tesla Motors](#) announced a new model [Roadster](#).

Systems must identify the textual spans which identify entities in text and under some task formulations additionally categorize the type of entity being mentioned — e.g. Tesla Motors represents a company and Roadster represents a product. This task was first formalized as part of the Message Understanding Conference (MUC) run by the Defense Advanced Research Projects Agency (DARPA) from 1987 to 1997. Starting with MUC-6 (Grishman and Sundheim, 1996), the conference hosted a shared task which required systems to identify and categorize mentions of entities, temporal and numerical expressions. Here entity mentions were to be categorized into coarse-grained types of Person (PER), Location (LOC) and Organization (ORG). While the MUC tasks concentrated on English language newswire text, subsequent tasks such as the Multilingual Entity Task (MET) (Merchant et al., 1996) and Conference on Natural Language Learning (CoNLL) tasks (Tjong Kim Sang, 2002; Tjong Kim Sang and Meulder, 2003) extended this evaluation to languages other than English. The CoNLL tasks also extended entity categories to include a catch-all miscellaneous (MISC) type for named entities outside the PER, LOC and ORG types.

While early approaches to NER were predominately driven by hand-crafted rules and heuristics, statistical and machine learning driven approaches have gradually

overtaken the best results (Nadeau and Sekine, 2007). Given that entity names play the linguistic role of identifiers in text, prior knowledge of a given name is often decisive in resolving mention spans where context alone provides little evidence — e.g. at sentence start where all tokens are capitalized. External knowledge in the form of training documents annotated with NE mentions and entity name gazetteers are therefor critical to high-performance NER (Ratinov and Roth, 2009).

Many approaches to automatically building gazetteers and producing these lexical resources have been proposed. Fundamental work in this area followed the bootstrapped pattern induction approach of Riloff and Jones (1999), where a small set of seed entities is used to identify textual patterns for entity references, which in turn produce more seed entities. This approach has been successfully applied to web documents to retrieve author and book titles (Brin, 1999) and build general dictionaries of named entities (Etzioni et al., 2005) from the web. While bootstrapped approaches only require a small set of seed names and unlabeled text to work, the quality of the generated results is often variable and has had little impact on NER systems. A promising alternative to bootstrapping is to leverage high-quality resources from a human-curated KB like Wikipedia to build large-scale gazetteers (Toral and Munoz, 2006; Kazama and Torisawa, 2007; Richman and Schone, 2008) or directly induce NER training data (Nothman et al., 2013). In both cases, these systems make use of human annotated inter-article links between pages to infer the presence of NE mentions, though the generalization of this idea to include links from the broader web remains to be explored.

2.2 Disambiguating entity mentions

NER incorporates both mention detection and entity type classification. However, the entity types assigned in standard NER are typically course-grained, high level and hard-coded; covering either the most prominent types, or types of interest in some specific

domain. While this kind of labeling is often sufficient, it doesn't resolve the fine-grained semantics of the specific entity being mentioned. For problems in information retrieval, question answering and knowledge base population, systems must both detect entity mentions and resolve ambiguity in the surface form used. Given that entities may go by multiple names and names may be shared by multiple entities, this remains a challenging task for automated systems.

To resolve mention ambiguity, coreferent entity mentions may either be tied to each other, or to corresponding nodes in some external KB. We discuss these two approaches to mention disambiguation in the following sections.

2.2.1 Coreference clustering

Coreference clustering resolves mention ambiguity by grouping together mentions of the same entity. This problem has been studied distinctly both within documents and across documents in a corpus.

The in-document version of this task, Coreference Resolution — seeks to group nominal and pronoun references into chains which refer to a common antecedent in the document. For example, while the first reference to an entity is commonly made by name, subsequent references may be indirect:

(7) Today [Tesla Motors](#) announced a new model Roadster.

[The company](#) expects to begin production in 2020.

Here we seek to cluster together both named references "Tesla Motors" and nominal "the company" for the same entity. Early approaches to this task focused on classifying coreference between individual mention pairs, then iteratively aggregating these decisions over a document to form full coreference chains. To make each pairwise coreference decision, systems have applied both supervised learning over mention pair features (Soon et al., 2001) and explicit syntactic and semantic constraints (Haghighi and Klein, 2009). Recent work integrates global consistency features to improve local coreference

decisions. For example, the mention-cluster classification framework (Haghighi and Klein, 2010) incorporates cluster level features which allow individual mentions to be compared to preceding, partially formed chains of mentions. While shallow linguistic features provide a very strong baseline, integration of features derived from large-scale external knowledge resources has been shown to improve performance (Ponzetto and Strube, 2006; Ratnov and Roth, 2012).

World knowledge features help to close the gap between humans and automated systems for non-trivial coreference decisions which rely on a reader’s common sense and prior knowledge. For example, specific knowledge that Tesla is a car company and not some other entity type can help reconcile the mentions considered in Example 7. Rahman and Ng (2011) explore the impact of world knowledge features in coreference systems, finding that incorporating external resources from structured KBs (YAGO and FrameNet), coreference annotated documents and even unannotated corpora all improve upon a baseline of linguistic features alone. Noise in web resources can however negate these performance gains without extensive preprocessing and filtering (Uryupina et al., 2011).

Cross Document Coreference Resolution (CDCR) identifies coreferent entity mentions across documents in a corpus. In contrast to Coreference Resolution, this task typically concentrates on proper noun references, either excluding or taking as given other in-document anaphora. Wacholder et al. (1997) first explored this disambiguation task over named entity mentions in Wall Street Journal articles. Their system heuristically scores mentions within a document to identify the least ambiguous name, then uses this as a canonical reference to align with other chains in the corpus. Comparisons are further constrained by types associated with each name in a precompiled gazetteer. Bagga and Baldwin (1998) are the first to utilize the textual context surrounding an entity mention to help resolve ambiguity. Their system groups in-document mentions into coreference chains, then computes weighted term-frequency vectors over the constituent sentences of each chain. Chains are then iteratively clustered across documents

using a threshold on their cosine similarity. Subsequent work has progressed in three main directions - bigger and better corpora for evaluation, richer context modelling and more robust and efficient clustering models. Gooi and Allan (2004) build a corpus of 25K person name mentions over New York Times articles and demonstrate the improvements from Hierarchical Agglomerative Clustering (HAC) in comparison to Bagga's iterative pairwise method. Rao et al. (2010) combine both orthographic name similarity and Latent Dirichlet Allocation (LDA) context vector representations to compute cluster similarity. While all entity types are important in general purpose CDCR, person names present a particularly challenging sub-problem which has attracted focused research. Mann and Yarowsky (2003) develop an unsupervised clustering system over features derived from extracted biographical facts for cross-document person coreference. They utilize the bootstrapped pattern induction of Ravichandran and Hovy (2002) to extract biographical facts such as birth place and occupation and show improved clustering results over basic textual context features alone. To evaluate their system, they construct a coreference corpus by searching the web for notable person names to retrieve document sets with little ambiguity, then inject ambiguity by randomly mixing pairs of retrieved documents for each entity and replacing their name references with a dummy pseudoname. This formulation of the person search problem later became the basis of the Web Person Search (WePS) task (Artiles et al., 2005) and a series of shared tasks at the Semantic Evaluation (SemEval) workshop (Artiles et al., 2007, 2009, 2010). WePS is a query driven formulation of the person name disambiguation task. Systems are given a set of search results for an ambiguous person name and must cluster coreferent results. Later versions of this task also evaluate the ability of systems to extract entity attributes from each cluster of retrieved documents.

Given the large-scale redundancy and variation in entity coverage across news, wikis, blogs and other online resources - the web presents a huge resource for general CDCR evaluation. Manual annotation of coreference amongst web documents is however a laborious and expensive task. Singh et al. (2011) automatically construct

a corpus of 1.5M disambiguated entity mentions from web pages with links to Wikipedia. They utilize this dataset to evaluate efficient large-scale CDCR using distributed hierarchical clustering. Web links and the semantic relationship they entail between entity references are a valuable resource for CDCR evaluation, though the utility of links to resources beyond Wikipedia remains to be explored.

2.2.2 Entity linking

Entity linking (EL) grounds ambiguous mentions in text to their correspond node in a structured KB. In contrast to coreference clustering, EL systems start with some knowledge of the entity set to be linked. Entities which don't exist in the KB at runtime are typically designated as NIL. This allows EL systems to take advantage of a potentially rich structured knowledge representation for candidate entities in the target KB when making disambiguation decisions. This integration of structured knowledge is the primary differentiator between the EL task setup and CDCR with partially formed clusters. In practice, EL systems often take advantage of many of the same features when resolving ambiguous textual mentions.

Work on entity linking has primarily targeted Wikipedia as a reference KB. Wikipedia is a large-scale, crowd-sourced encyclopedia with good coverage of notable entities - making it a natural target for evaluations over news articles and discussion on the web. Bunescu and Paşca (2006) first developed the EL framework in terms of two component tasks of entity detection and disambiguation. To detect entities, they look-up proper names references in a name-entity dictionary derived from Wikipedia page titles, redirects and disambiguation page entries. For detected names which align with multiple entities, they utilize a supervised Support Vector Machine (SVM) ranker which scores feature vectors capturing correlation between the context of a query mention and each candidate entity. They find that modelling category-term combinations improves upon a baseline of TF-IDF weighted term vectors alone, though they must constrain the set of categories to avoid generating an intractably large

feature space. Beyond textual context, Milne and Witten (2008) incorporate features which model both the popularity of a candidate entity and its relatedness to other unambiguous entities mentioned in the same document. Inter-entity relatedness has proven to be a rich source of evidence for named entity disambiguation. Many systems have since employed graph-based collective disambiguation methods (Han et al., 2011; Hoffart et al., 2011) and Personalized Page Rank (PPR) (Alhelbawy and Gaizauskas, 2014; Pershina et al., 2015) to take advantage of inter-entity dependence. While the disambiguation problem has attracted much attention, robust entity detection and candidate look-up remains a challenging component of the EL task which often limits overall performance (Hachey et al., 2013). Systems which solve this task jointly with the entity disambiguation problem tend to yield the best results (Cucerzan, 2007), especially when linking noisy sources in the social media domain like Twitter (Meij et al., 2012a; Guo et al., 2013). More recently, neural models of entity recognition and disambiguation have featured prominently. These models offer richer context modelling and joint learning of representations for entities, words and relations between the two. He et al. (2013a) demonstrate a competitive disambiguation system using simple document level representations learnt with Stacked Denoising Autoencoders (SDA). Subsequent work investigates the joint embedding of entities and words (Yamada et al., 2016) and attention over local context with differentiable collective disambiguation (Ganea and Hofmann, 2017).

2.3 Stores of entity knowledge

Stored knowledge sourced from the web or otherwise is central to entity disambiguation. Various projects have tried to build upon and extend Wikipedia coverage for EL and other IE tasks. DBpedia (Auer et al., 2007) extracts structured information from Wikipedia and YAGO (Hoffart et al., 2013) extends coverage by integrating knowledge from sources like GeoNames and WordNet. Freebase (Bollacker et al., 2008) in particular

attracted attention as a broad target for entity disambiguation (Zheng et al., 2012a) and became the source for the Text Analysis Conference (TAC) shared task on EL in 2015 (Ji et al., 2015). Originally developed by MetaWeb and later acquired by Google, Freebase grew to cover almost 40M entities and 2B facts before being discontinued in favour of the closed-access Google Knowledge Graph. Since then, contributors have tried to merge the remaining open-access portion of Freebase into Wikidata – the centralized store of structured knowledge across multiple language Wikipedias (Vrandečić and Krötzsch, 2014). Wikidata presents one of the best targets for entity knowledge applications — providing open access to a dataset of over 45 million items (most but not all named entities¹) and almost half a billion statements (e.g. facts) about those items. Moreover, up-to-date snapshots of this data are published on a weekly basis through JSON encoded data dumps². For applications which seek to explore the mapping between structured KB representations and unstructured text, Wikidata also provides a direct alignment between KB facts and natural language descriptions of an entity across multiple language Wikipedias.

Given the wide variety of knowledge resources and entity coverage available across structured KBs, much work has explored the task of aligning and consolidating knowledge across these resources. Tasks such as record linkage (Fellegi and Sunter, 1969; Xu et al., 2013) and entity alignment (Hao Zhu, 2017) match instances across distinct KBs by learning to align equivalent structured relations across distinct schema. In the web domain, Semantic Web (Berners-Lee et al., 2001) and linked open data (Bizer et al., 2008) initiatives³ address a similar goal — attempting to build a globally distributed store of machine-readable knowledge. Here ontology matching (Euzenat and Shvaiko, 2007) systems address the analogous task of automatically aligning concepts across distinct ontologies and initiatives such as `schema.org`⁴ attempt to address the alignment problem by promoting a common set of schema for knowledge description. In

¹<https://www.wikidata.org/wiki/Wikidata:Statistics>

²<https://dumps.wikimedia.org/wikidatawiki/entities/>

³<http://linkeddata.org/>

⁴<https://schema.org>

practise, formal specification and encoding of knowledge has proven to be challenging (Shipman and Marshall, 1999) and widespread adoption of semantic web technology has been slow (Mika, 2017). As dedicated entity ontologies continue to slowly grow and improve coverage, alternative approaches which aggregate entity knowledge across KBs (Han and Zhao, 2010; Sil et al., 2012) and down the long-tail of entity notability in social media (Jin et al., 2014) are a promising alternative for web-scale entity coverage. However, as information extraction systems improve, far more resources originally published for human consumption may be utilized by automated systems — reducing the demand for manually curated machine readable knowledge.

Connections between unstructured resources on the web and structured KBs present an opportunity for the development of systems addressing this task. Existing work in this domain includes the automatic annotation of web corpora with links back to a structured KB (Gabrilovich et al., 2013), and systems which seek to discover web pages associated with specific KB entries (Hachenberg and Gottron, 2012). Our work in this thesis focuses on inbound links to structured or semi-structured web resources from documents on the web — e.g. news articles, blog posts, forum discussion or any other natural language encoded entity description with outbound links back to a more structured KB or KB-like web endpoint. As part of their investigation of large-scale CDCR, Singh et al. (2012) develop and distribute the Wikilinks corpus which contains over 40M web mentions of nearly 3M Wikipedia entities. This dataset is central to our exploration of inlink driven entity disambiguation in Chapter 3 and inspires development of generalized inlink-driven entity knowledge resources in subsequent chapters. This data distills what is predominately human annotated entity coreference across structured and unstructured resources — enabling the development of automated systems for mining and aligning entity references on the web.

2.4 Extracting structured knowledge

Entity detection and disambiguation help resolve the question of who or what is being talked about in natural language — but do not account for specifically what is being said. One way of representing this knowledge is through relations, i.e. structured triples of subject, predicate and object which represent the facts conveyed through natural language. Relation Extraction (RE) is the task of identifying and classifying the relations expressed between entities in unstructured text. For example, given a sentence such as "Elon Musk is the CEO of Tesla Motors." we may aim to identify the `chief_executive_officer` relation between the Elon Musk and Tesla Motors entities. Extracted relations are directly applicable to language understanding problems like Question Answering (QA) where relational triples stored in a KB may be queried to resolve questions.

Early work on RE centered around the MUC-7 shared task (1997) and ACE evaluations (2000-2007). Systems primarily utilized handcrafted rules (e.g. Aone et al. (1998)) and supervised learning (e.g. Zelenko et al. (2003)) to identify relations. Rule based approaches can recover relations with high-precision over short textual spans, but do not generalize well to longer more diverse sequences. Bootstrapped pattern learning (Brin, 1999; Agichtein and Gravano, 2000) can help address this recall issue, but is limited by semantic drift over repeated iterations without human supervision (Curran et al., 2007). Supervised approaches can generalize over textual forms by incorporating higher level linguistic features, but suffer from a scarcity of annotated training data relative to the huge diversity in how relations may be expressed in text. Subsequent work has explored semi and self-supervised approaches which augment the learning process through the incorporation of external resources and unlabeled text. Wu and Weld (2007) first describe a mechanism for training a supervised relation classifier over unlabeled text on their fact of info-box attribution extraction with Wikipedia articles. Subsequently, Mintz et al. (2009) formalize this approach as the distant supervision

framework for relation extraction. To generate training instances, they each search an unannotated corpus for sentences containing pairs of entities that appear in known relations, then label each sentence as if they express those relations. This silver-standard annotation helps mitigate the sparsity problems inherent to small hand-labeled corpora and efficiently leverages existing structured knowledge. This process is not however without its drawbacks. Missing KB relations can induce false negatives and sentences which don't express the same relation found in the KB produce false positives amongst recovered entity pairs (Roth et al., 2013). Moreover, the potential for multiple relations between an entity pair is not accounted for. Surdeanu et al. (2012) address these issues in part by explicitly modelling the multi-instance, multi-label nature of the learning processing in distant supervision. In practice, facts about an entity may be expressed across multiple sentences and throughout multiple documents in a corpus. During aggregation these facts may be redundant, complementary or even contradictory. Slot Filling (SF) is a reformulation of the base RE task which directly models the end-to-end extraction problem over multiple documents and entities. SF has been a focus of the TAC KBP track since 2009 (McNamee et al., 2009) and is generally considered a challenging task for automated systems. While the dominant approach to SF has long been a pipeline of tasks involving search, entity recognition, disambiguation, coreference resolution and finally candidate fill extraction and ranking — recent work has begun to replace aspects of this pipeline with neural networks and end-to-end learning (Nguyen and Grishman, 2015; Adel et al., 2016; Huang et al., 2017).

Incorporation of an existing KB as a source of supervision imposes a schema on the types of relations that may be extracted. Open Information Extraction (OpenIE) bypasses this requirement by directly learning relation types from the target corpus. In this framework, relations are loosely defined in terms of the textual spans or phrases which denote a relationship. Following the example above: "Elon Musk is the CEO of Tesla Motors"; we may equally well encode the `chief_executive_officer` relation between Tesla and Musk as (Elon Musk, is the CEO, Tesla Motors). Banko et al. (2007) first explore

this approach to RE with their **TEXTRUNNER** system. They first train a Naive Bayes relation tuple extractor over a small corpus through self-supervision using heuristic dependency parse constraints. They then apply this classifier to extract candidate triples over a larger corpus and assign probabilities to extracted relations based on a model of redundancy across sentences in the corpus. They apply this system to a corpus of 9M web documents and are able to efficiently extract millions of relations across a broad set of types, though it is difficult to completely account for redundant relations under this evaluation. Subsequent work builds upon this approach to improve the precision and recall of extractors. For example, the **REVERB** system (Fader et al., 2011) which introduces syntactic and lexical constraints that significantly reduce the number of incoherent and uninformative extracts and **OLLIE** (Mausam et al., 2012) which extends extraction beyond relations mediated by verbs. While relations expressed under in this form are not bound by a fixed schema, many downstream applications require a fixed and canonicalized set of relations. For example, if we wish to find all company CEOs in a KB through structured search, we must account for the alternative ways this relation may be expressed under an open schema. Universal schema (Riedel et al., 2013) resolve this problem by learning to align equivalent relations express across distinct schema — in the case of OpenIE, mapping relations expressed under the language itself unto a fixed and finite set of equivalent KB relations.

We have primarily discussed systems addressing the task of structured information extraction from unstructured natural language text. When working with text on the web, systems may leverage far more information than the textual content of a page. Craven et al. (1998) describe a system for constructing web-derived knowledge bases over a predefined schema by classifying pages which represent entities or express relationships. They learn to predict pages which belong to classes within an ontology and exploit links between classes to infer relationships between identified entities. Systems may also extract information from HTML markup of list and table elements (Cafarella et al., 2008) and extract responses to queries made via form elements which

exploit the structure of the deep web (Bergman, 2001; Madhavan et al., 2008). These approaches may be combined with traditional text-based bootstrapped pattern induction (Etzioni et al., 2005) and OpenIE relation extraction systems (Cafarella et al., 2009). Carlson et al. (2010) develop the Never Ending Language Learner (NELL) system which combines textual patterns and structured extractors over lists and tables on the web with a supervised regression model to classify the likelihood of candidate facts. Facts with a probability over the belief threshold are included into the KB and integrated into subsequent iterations of rule discovery and retraining. More recently, systems integrating information from both previous extractions and existing structured KB resources have been used to improve subsequent web extractions (Lao et al., 2012; Dong et al., 2014).

The web has long been both a source and target for IE systems. Systems may leverage both the structure of page content and the links between pages to sample training data for machine learning, or utilize these structured as features at run-time to better extract information. Our work represents a continuation of this trend, focusing first on the implied semantics of web links to KB targets and later on how structured information may be extracted from entity mentions identified this way.

2.5 Learned representations

IE requires a deep understanding of natural language; a medium which is often noisy, inconsistent and ambiguous. Up to this point, we have described the dominant approach to IE as a pipeline of isolated and contingent tasks. Systems address each task in turn, often through the integration of features which leverage specific linguistic insights. While successful, this approach requires time-intensive feature engineering and tends to yield fragile systems which generalize poorly to new domains and tasks. Deep learning is an approach to machine learning where feature representations for a domain are learnt directly from data. This approach can negate the need for

hand-crafted features and offer a mechanism for building compact, reusable language representations which embed features relevant across a variety of tasks. To achieve these characteristics, deep learning systems often rely upon a combination of large labeled and unlabeled data sources and extensive computation. In recent years, deep learning models have achieved well-publicized results in fields like speech (Graves et al., 2013), vision (Krizhevsky et al., 2012) and game playing (Mnih et al., 2015) without leaning on significant prior knowledge of the target domain. This is often accomplished by replacing pipelines of traditional systems with a single large Neural Network (NN) trained to optimize the end-to-end task objective.

While learned feature representations are clearly a powerful tool applicable to problems across multiple domains, deep learning alone is no panacea for artificial intelligence tasks. In practice, much of the effort previously applied to feature engineering is now applied to the engineering of NN model architecture. Notably, convolutional networks (LeCun and Bengio, 1998) which model spacial patterns, recurrent network (Elman, 1990; Hochreiter and Schmidhuber, 1997) which model sequential data and recursive models (Goller and Küchler, 1996; Socher et al., 2011) which enable end-to-end learning over nested structures. While the core ideas and models at the root of deep learning have been around for decades, increased research attention, larger datasets and increased access to high-performance computation (e.g GPUs, TPUs⁵) and software tools (e.g. TensorFlow⁶, Torch⁷) has increased the applicability of these models across domains.

In this section we describe work adapting neural network models to the natural language domain. We address the representation of words, entities and relations and describe how large-scale data from the web and KBs may be leveraged by these models.

⁵<https://cloud.google.com/tpu>

⁶<https://www.tensorflow.org/>

⁷<http://pytorch.org>

2.5.1 Text representation

One of the fundamental challenges in IE, and Artificial Intelligence in general, is how best to internally represent knowledge. Distributed representations (Hinton, 1986) encode information using dense, continuous valued vectors. Components of the vector may be understood as factors describing a concept. This compositional format is desirable – given small perturbations in the vector retain similar *meanings*, it is inherently robust to noise and can generalize well. Despite this appeal, sparse symbolic language representations have historically dominated the field of NLP. While neural models have been around for decades, their performance has traditionally been limited by the difficulty of gradient descent learning in deep, non-linear networks. In 2006, work demonstrating that greedy, layer-wise training of deep networks was possible using unlabeled data reignited interest in the field (Hinton and Salakhutdinov, 2006; Bengio et al., 2007). Taking this approach, weights for each layer of the network were trained to represent and reproduce input from the layer below. This process progressively develops higher level representations of the input data – eventually, identifying and disentangling the underlying explanatory factors behind the input and improving performance on subsequent supervised prediction tasks (Bengio et al., 2013).

In NLP, the idea of learning distributed representations for words through the language modeling task was first introduced by Bengio et al. (2003). Their model used a single layer neural network to predict the probability of the next word in a sequence, given the learned representations for a set of context words. This approach exploits the distributional hypothesis (Harris, 1954); deriving word meaning from the fact that similar words tend to appear in similar contexts. This model was extended by Collobert and Weston (2008) to make use of convolutional, multi-layer networks. In addition to unsupervised language modeling, the network was applied to a variety of standard NLP tasks covering both syntax (e.g. POS tagging) and semantics (e.g. synonym detection).

This model was shown to perform near or better than state-of-the-art on each task, with the best performing systems having been trained jointly across all tasks.

One advantage of word representations trained in this manner is their ability to share the statistical strength of representations derived from a large unlabeled corpus with supervised tasks that would otherwise rely on a small labeled dataset. Pre-trained word representations such as Word2Vec (Mikolov et al., 2013a), GloVe (Pennington et al., 2014) and FastText (Bojanowski et al., 2017) embeddings have been shown to increase performance on downstream tasks both as features (Turian et al., 2010) and initializations (Socher et al., 2013; Kumar et al., 2016) for task specific representations. This ability to leverage unlabeled data and transfer knowledge across tasks enables the integration of large-scale unstructured resources of the web into NLP tasks. For example, open-access embedding models pretrained on Google News⁸, CommonCrawl⁹ and ClueWeb (Zamani and Croft, 2017).

2.5.2 Representing entities and relations

Learned representations that capture the general or task specific semantic attributes of words and phrases are now common throughout the field of NLP. For applications in the information extraction domain, it is natural to consider directly learning representations for higher order constructs such as entities and relations.

Entity representation has been explored through simple extensions of the word vector model to collocated multi-word phrases (Mikolov et al., 2013b). This model yields useful entity representations, embedding semantically similar entities close together in word-vector space. For example, the embedding for *Paris* will be close to the embedding of *Madrid*. Moreover, directions within the induced vector space have also been shown to encode relations between entities. These relation vectors enable a kind of analogical reasoning, e.g. the vector returned by an operation such as:

⁸<https://code.google.com/archive/p/word2vec/>

⁹<https://nlp.stanford.edu/projects/glove/>

Paris – France + Spain will be close in vector-space to the embedding for Madrid. One limitation of this approach is that embeddings are tied to specific linguistic surface forms which represent a given entity name, rather than a specific entity itself. Subsequent work addresses this issue by inducing multiple prototype word representations (Reisinger and Mooney, 2010) and broader document context (Huang et al., 2012; Kusner et al., 2015) to resolve multiple underlying meanings for a given surface form.

While the ability to extract useful semantic representations from an unsupervised language modelling objective is compelling, systems may also learn entity and relation representations from structured sources or as a by-product of addressing some other extrinsic task objective. One such example is the task of Knowledge Base Completion (KBC), where systems seek to predict new relations for KB entities given those already populated in the KB. Bordes et al. (2011) develop a structured embedding model for KBC by learning to score in-KB relation triples above randomly generated out-of-KB triples. In this model, triples are scored by measuring the euclidean distance between learned representations for the subject and object entities after projection via a learned relation embedding. This model can then be adapted to score and estimate the likelihood of previously unseen triples. The utilization of a dense, distributed knowledge embedding may be contrasted with prior approaches to relation inference which utilize classical symbolic reasoning (Lenat, 1995; Kok and Domingos, 2007) or sparse matrix factorization (Singh and Gordon, 2008; Riedel et al., 2013) to predict missing KB triples. Subsequent models have enabled richer modelling of the interaction between entities and relations (Socher et al., 2013; Wang et al., 2014; Lin et al., 2015) to improve KBC predictions and improve upon relation extraction performance by training on a shared objective (Weston et al., 2013; Toutanova et al., 2015).

2.6 Summary

In this section we give a broad background on information extraction systems and highlight the role the web resources have played in the development of this work. The web is an appealing target for information extraction systems. It is rich in terms of the breadth and depth of content available; cheap in that knowledge is often transcribed as a by-product of third party activity; and easy wherever generated content follows patterns which may later be exploited by automated systems. Large-scale data sets extracted from web resources with weak or even no supervision are also a good fit for data-hungry deep learning models which can scale up model capacity to take advantage of larger datasets. Models trained on this data may then transfer acquired knowledge across tasks by exporting reusable embedding representations or directly modelling a joint multi-task objective.

In subsequent chapters we address a variety of tasks in turn — first building out a framework for mining and aligning entity references from the web and later investigating the transformation of extracted information using deep neural networks and end-to-end learning. While this chapter helps set the scene for work considered in later chapters, we will subsequently revisit background for the specific experiments described therein.

3 Entity Disambiguation with Web Links

The litmus test for whether you are a competent forecaster is if more information makes your predictions better.

Nate Silver

Ambiguity is a key challenge in many language understanding problems. For applications in IE, resolving entity ambiguity is a fundamental prerequisite to subsequent extraction tasks. If a system cannot robustly identify specifically who or what is being referenced in text, extracted knowledge is not grounded to objects and concepts from the real world. Named entity disambiguation (NED) systems resolve ambiguity by modelling the way in which entities are mentioned in text. These models are traditionally derived from aggregated information about an entity in a structured KB. However, linked data from the web can provide an equivalent source of knowledge. Whenever text on the web is annotated with links to a page representing an entity, we may potentially leverage the content and context of those links to extract information about the entity. For NED, links are directly applicable as they represent samples of natural entity mentions in text. However, datasets derived from the web can be noisy. Content creators are not bound by the same quality constraints and accuracy standards as contributors to a reviewed encyclopedic KB. Nor do they approach content creation with the same motivations and intent. Web links also lack analogues of standard KB

structure commonly used for disambiguation, e.g. category annotations and relational knowledge.

In this chapter, we investigate whether disambiguation models derived from web links can be used to augment or even completely replace knowledge sourced from a traditional structured KB. We develop this framework by focusing on Wikipedia as a target KB given its prominence as a benchmark in NED. In later chapters, we build upon this groundwork to generalize link-driven disambiguation to multiple web KBs.

Contributions include: (1) a benchmark linker that instantiates entity prior probabilities, entity given name probabilities, entity context models, and efficient entity coherence models from Wikipedia-derived data sets; (2) an alternative linker that derives the same model using only alternative names and web pages that link to Wikipedia; (3) detailed development experiments, including analysis and profiling of Web link data, and a comparison of link and Wikipedia-derived models. Experiments detailed in this chapter were first described in Chisholm and Hachey (2015). Applications of this work to semantic indexing is described in Cadilhac et al. (2015). Our linking systems is available under an open-source licence at: github.com/wikilinks/nel.

3.1 Introduction

Wikipedia and related semantic resources, e.g. Freebase, DBpedia, YAGO — have emerged as general repositories of notable entities. The availability of Wikipedia, in particular, has driven work on NED, knowledge base population (KBP), and semantic search. This literature demonstrates that the rich structure of Wikipedia— redirect pages, article text, inter-article links, categories — delivers disambiguation accuracy above 85% on newswire (He et al., 2013b; Alhelbawy and Gaizauskas, 2014). But what disambiguation accuracy can we expect in the absence of Wikipedia’s curated structure?

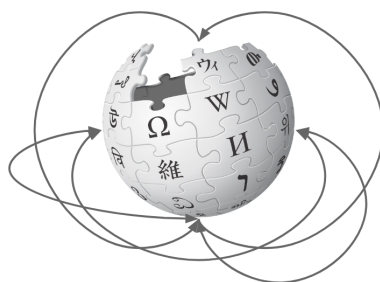


Figure 3.1: Inter-article links across pages from Wikipedia.

Web links provide much of the same information as Wikipedia inter-article links. Anchors are used to derive alternative names and conditional probabilities of entities given names; in-link counts are used to derive a simple entity popularity measure; the text surrounding a link is used to derive textual context models; and overlap of in-link sources is used to derive entity co-occurrence models. Figures 3.1 and 3.2 depict the comparable link structure of inter-article and external web link sources respectively.

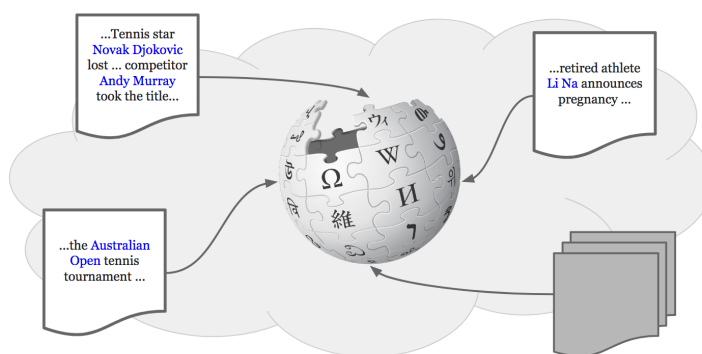


Figure 3.2: Inlinks to Wikipedia from external sources.

If this source of entity knowledge is proven to be effective, it is an appealing alternative to traditional knowledge sources. Web links are an incidental source of entity annotation — typically generated as a byproduct of third party web publishing activity. Links also provide a more natural and diverse distribution of coverage in contrast to Wikipedia which is curated by a comparatively small group of editors. Link driven disambiguation also generalizes effortlessly across diverse and distinct KB schema – an important characteristic we exploit in later chapters. On the other hand, web links lack

analogues for Wikipedia structure commonly used in disambiguation, e.g., categories, info-box fields and encyclopedic descriptions. Moreover, Wikipedia’s editors maintain a clean and correct knowledge source while web links are a potentially far noisier source of annotation.

We identify a set of general disambiguation features which can be derived from both Wikipedia and web link sources. We then seek to answer three key research questions. First, how does disambiguation performance compare for different features across each source? Next, can feature combinations for web link sources alone compete with those derived from Wikipedia? Finally, are features combinations across sources complementary, and how do they compare with state of the art disambiguation results? We depict this unified view of web and Wikipedia disambiguation resources in Figure 3.3.

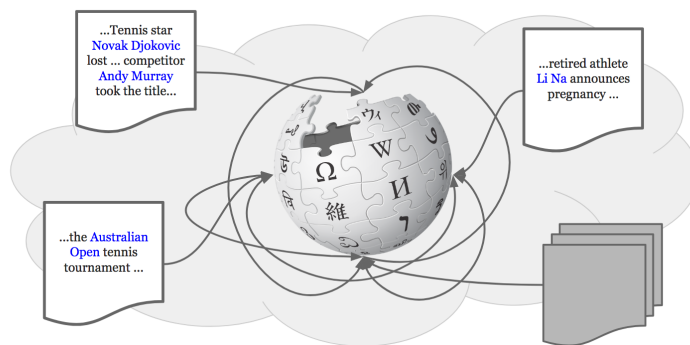


Figure 3.3: Combined Web and Wikipedia inlinks.

Results suggest that web link accuracy is at least 93% of a Wikipedia linker and that web links are complementary to Wikipedia, with the best scores coming from a combination which competes with state-of-the-art results for NED on newswire.

3.2 Related work

Thomas et al. (2014) describe a disambiguation approach that exploits news documents that have been curated by professional editors. Document level entity tags are exploited

to build textual mention context, assign weights to alternative names, and train a disambiguator. This leads to an estimated F score of 78.0 for end-to-end linking to a KB of 32,000 companies. The ability to replace a dedicated KB is compelling, though this approach requires a dedicated curation effort for published articles. Our work is similar, but we replace quality curated news text with web pages and explore a larger KB of more than four million entities. In place of document-level entity tags, hyperlinks pointing to Wikipedia articles are used to build context, name and coherence models. Li et al. (2013) explore a similar task setting for microblogs, where short mention contexts exacerbate sparsity problems for underdeveloped entities. They address the problem by building a topic model based on Wikipedia mention link contexts. A bootstrapping approach analogous to query expansion augments the model using web pages returned from the Google search API. Results suggest that the bootstrapping process is beneficial, improving performance from approximately 81% to 87% accuracy. We demonstrate that adding link data leads to similar improvements.

The cold start task of the Text Analysis Conference is also comparable.¹ It evaluates how well systems perform end-to-end NIL detection, clustering and slot filling. Input includes a large document collection and a slot filling schema. Systems return a KB derived from the document collection that conforms to the schema. The evaluation target is long-tail or local knowledge. The motivation is the same as our setting, but we focus on cold-start linking rather than end-to-end KB population.

Finally, recent work addresses linking without and beyond Wikipedia. Jin et al. (2014) describe an unsupervised system for linking to a person KB from a social networking site, and Shan et al. (2014) describe a general approach for arbitrary KBs. Nakashole et al. (2013) and Hoffart et al. (2014) add a temporal dimension to NIL detection by focusing on discovering and typing emerging entities.

¹<http://www.nist.gov/tac/2014/KBP/ColdStart/guidelines.html>

3.3 Terminology

In this section we provide a reference for common terms used throughout this chapter.

- **Link** — a highlighted span of text used to reference another document on the web. Clicking on a link navigates a user to the targeted page.
- **Target** — the web page or document addressed by a link.
- **Anchor** — the textual span highlighted within a document by a link.
- **Named entity** — a distinct and independent person, place, object or thing from the world which may be referenced by name.
- **Entity page** — a web page which uniquely describes or references a named entity. For example, an article describing an entity in Wikipedia.
- **Inlinks** — links into a page from other sources on the web.
- **Article** — the textual content of a entity page. Other page content (e.g. images, info-box, tables) is not considered.
- **Mention** — an instance where an entity is referenced in text. In this chapter we consider inlinks to an entity page to represent mentions of that entity in text. In addition, we assume the anchor for these links to represent a valid entity alias.
- **Alias** — an alternative name for an entity. For example, the American rapper Marshal Mathers is also known as Eminem and Slim Shady. In this chapter we assume the anchor for an inlink to an entity page may represent an alias for the entity.

3.4 Tasks

Two evaluations in particular have driven comparative work on NED: the TAC KBP shared tasks and the YAGO annotation of CoNLL 2003 NER data. We describe these tasks and their respective evaluation setup. A brief survey of results outlines the kind of performance we hope to achieve with link data. For task history, we suggest Hachey et al. (2013) and Shen et al. (2015). For an evaluation survey, see Hachey et al. (2014) and for a review of other prominent EL benchmarks and datasets (e.g. ACE, MSNBC, ACQUAINT, IITB) see Ling et al. (2015).

Our evaluation setup follows He et al. (2013b) for comparability to their state-of-the-art disambiguation results across CoNLL and TAC data. This configuration does not replicate the end-to-end entity linking (EL) task which combines both entity recognition and disambiguation. Instead we take as input a standard set of mention annotations and evaluate disambiguation performance in isolation. Table 3.1 summarises the data sets used. Columns correspond to number of documents ($|\mathcal{D}|$), number of entities ($|\mathcal{E}|$), number of mentions ($|\mathcal{M}|$), and number of non-NIL mentions ($|\mathcal{M}_{kb}|$). The non-NIL mention number represents the set used for evaluation in the disambiguation experiments here. The table also includes average and standard deviation of the candidate set cardinality over \mathcal{M}_{kb} ($\langle C \rangle$) and the percentage of mentions in \mathcal{M}_{kb} where the correct resolution is in the candidate set (R_C). The last column (SOA) gives the state-of-the-art score from the literature. Numbers are discussed below.

3.4.1 CoNLL

CoNLL is a corpus of RCV1 newswire annotated for whole-document named entity recognition and disambiguation (Hoffart et al., 2011). CoNLL is public, free and much larger than most entity annotation data sets, making it an excellent evaluation target. It is based on the widely used NER data from the CoNLL 2003 shared task (Tjong Kim Sang and Meulder, 2003), building disambiguation on ground truth mentions. Training

Data set	$ \mathcal{D} $	$ \mathcal{E} $	$ \mathcal{M} $	$ \mathcal{M}_{kb} $ (%)	$\langle C \rangle$	(σ)	R_C	SOA
CoNLL train	945	4,080	23,396	18,505 (79)	69 (194)		100	NA
CoNLL dev	216	1,644	5,917	4,791 (80)	73 (194)		100	79.7
CoNLL test	231	1,537	5,616	4,485 (80)	73 (171)		100	87.6
TAC train	1,040	456	1,500	1,070 (71)	23 (28)		94.4	NA
TAC test	1,012	387	2,250	1,017 (45)	24 (30)		88.5	81.0

Table 3.1: Data sets for disambiguation tasks addressed here. Statistics are described in Section 3.4.

and development splits comprise 1,162 stories from 22-31 August 1996 and the held-out test split comprises 231 stories from 6-7 December 1996.

The standard evaluation measure is *precision@1* ($p@1$) – the percentage of linkable mentions for which the system ranks the correct entity first (Hoffart et al., 2011). Linkable is defined as ground truth mentions for which the correct entity is a member of the candidate set. This factors out errors due to mention detection, coreference handling, and candidate generation, isolating the performance of the proposed ranking models. For comparability, we use Hoffart et al.’s YAGO *means* relations for candidate generation. These alternative names are harvested from Wikipedia disambiguation pages, redirects and inter-article links. In the Hoffart et al. setting, candidate recall is 100%.

There are several key benchmark results for the CoNLL data set. Hoffart et al. (2011) define the task settings and report the first results. They employ a global graph-based coherence algorithm, leading to a score of 82.5. He et al. (2013b) present the most comparable approach. Using deep neural networks, they learn entity representations based on similarity between link contexts and article text in Wikipedia. They report performance of 84.8 without collective inference, and 85.6 when integrating Han et al. (2011) coherence algorithm. Finally, Alhelbawy and Gaizauskas (2014) report the current best performance of 87.6 using a collective approach over a document-specific subgraph.

3.4.2 TAC 2010

Since 2009, the Text Analysis Conference (TAC) has hosted an annual EL shared task as part of its Knowledge Base Population track (KBP) (Ji and Grishman, 2011). Through 2013, the task is query-driven. Input includes a document and a name that appears in that document. Systems must output a KB identifier for each query, or NIL. The KB is derived from a subset of 818,741 Wikipedia articles. We use data from the 2010 shared task for several reasons. First, it facilitates comparison to current art. Second, it is a linking-only evaluation as opposed to linking plus NIL clustering. Finally, it includes comparable training and test data rather than relying on data from earlier years for training.

The TAC 2010 source collection includes news from various agencies and web log data. Training data includes a specially prepared set of 1,500 web queries. Test data includes 2,250 queries – 1,500 news and 750 web log uniformly distributed across person, organisation, and geo-political entities. Candidate generation here uses the DBpedia lexicalizations data set (Mendes et al., 2012), article titles, and redirect titles. We also add titles and redirects stripped of appositions indicated by a comma (e.g., *Montgomery, Alabama*) or opening round bracket (e.g., *Joe Morris (trumpeter)*). Candidate recall is 94.4 and 88.5 on the training and test sets – an upper limit on disambiguation accuracy.

Following He et al., we report KB accuracy (A_{kb}) - the percentage of correctly linked non-NIL mentions - to isolate disambiguation performance. Before evaluation, we map Wikipedia titles in our output to TACKB identifiers using the Dalton and Dietz (2013) alignment updated with Wikipedia redirects. To our knowledge, Cucerzan (2011) report the best A_{kb} of 87.3 for an end-to-end TAC entity linking system, while He et al. (2013b) report the best A_{kb} of 81.0 for a disambiguation-focused evaluation. There are a number of differences, e.g.: mention detection for coherence, coreference modelling, and substring matching in candidate generation. Analysis shows that these can have a large effect on system performance (Hachey et al., 2013; Piccinno and Ferragina, 2014).

We use He et al.’s setup to control for differences and for comparability to He et al.’s results.

3.5 Wikipedia benchmark models

A wide range of EL approaches have been proposed that take advantage of the clean, well-edited information in Wikipedia. These include entity prior models derived from popularity metrics; alias models derived from Wikipedia redirects, disambiguation pages and inter-article links; textual context models derived from Wikipedia article text; and entity coherence models derived from the Wikipedia inter-article link graph. We survey these models and describe a new benchmark linker that instantiates them from existing Wikipedia-derived data sets. For a more detailed survey of features in supervised systems, see Meij et al. (2012b) and Radford (2014).

Table 3.2 contains an overview of $p@1$ results for individual components on the CoNLL development data.

3.5.1 Entity prior

The simplest approach to entity disambiguation ranks candidate entities in terms of their popularity. For example, 0.000001% of inter-article links in Wikipedia point to Nikola Tesla, while 0.000008% point to Tesla Motors. An entity prior is used in generative models (Guo et al., 2009; Han and Sun, 2011) and in supervised systems that incorporate diverse features (Radford et al., 2012). We define the entity prior feature as the log-probability of a link pointing to entity e :

$$f_{prior}(e) = \log \frac{|\mathcal{I}_{*,e}|}{|\mathcal{I}_{*,*}|}$$

where $\mathcal{I}_{*,e} \in \mathcal{I}_{*,*}$ is the set of pages that link to entity e . We derive this from DBpedia’s Wikipedia Pagelinks data set, which contains the link graph between Wikipedia pages.² Missing values are replaced with a small default log probability of -20, which

²<http://wiki.dbpedia.org/Downloads>

Component	Articles	Mentions	Web links
f_{prior}	68.4	68.4	63.0
f_{name}	69.2	69.2	58.4
f_{bow}	50.6	55.8	62.2
f_{dbow}	49.9	51.2	54.0

Table 3.2: $p@1$ results for individual components on the CoNLL development data. The first two columns correspond to the Wikipedia models described in Section 3.5.3, one derived from article text and the other from mention contexts. The last column corresponds to the web link models described in Section 3.6.

works better than add-one smoothing in development experiments. On the CoNLL development data, entity prior alone achieves 68.4 $p@1$.

3.5.2 Name probability

Name probability models the relationship between a name and an entity. For example, 0.04% of links with the anchor text ‘Tesla’ point to Nikola Tesla, while 0.03% point to Tesla Motors. Name probability was introduced as an initial score in coherence-driven disambiguation (Milne and Witten, 2008), and is used in most state-of-the-art systems (Ferragina and Scaiella, 2010; Hoffart et al., 2011; Cucerzan, 2011; Radford et al., 2012). We define the name probability feature as the log-conditional probability of a name referring to an entity:

$$f_{name}(e, n) = \log \frac{|\mathcal{M}_{n,e}|}{|\mathcal{M}_{n,*}|}$$

where $\mathcal{M}_{n,e}$ is the set of mentions with name n that refer to entity e and $\mathcal{M}_{n,*}$ is all mentions with name n . We use existing conditional probability estimates from the DBpedia Lexicalizations data set (Mendes et al., 2012).² This derives mentions from Wikipedia inter-article links, where names come from anchor text and referent entities from link targets. Estimates for entities that have fewer than five incoming links

are discarded. We smooth these estimates using add-one smoothing. On the CoNLL development data, name probability alone achieves 69.2 $p@1$.

3.5.3 Textual context

Textual context goes beyond intrinsic entity and name popularity, providing a means to distinguish between entities based on the words with which they occur. For example, references to Tesla the car manufacturer appear in passages with words like ‘company’, ‘electric’, ‘vehicle’. References to the inventor appear with words like ‘engineer’, ‘ac’, ‘electrical’. Textual context was the primary component of the top system in the first TAC evaluation (Varma et al., 2009), and is a key component in recent art (Ratinov et al., 2011; Radford et al., 2012).

3.5.3.0.1 BOW context We model textual context as a weighted bag of words (BOW), specifically as a term vector \vec{t} containing TF-IDF weights:

$$tfidf(t, p) = \sqrt{f(t, p)} \cdot \log \left(\frac{|\mathcal{D}|}{|\{d \in \mathcal{D} | t \in d\}|} \right)$$

where t is a term, p is a passage of text, $f(t, p)$ is the term frequency of t in p , $|\mathcal{D}|$ is the total number of documents, and $\{d \in \mathcal{D} | t \in d\}$ is the set of documents containing t (Salton and Buckley, 1988). We derive the term frequency for an entity e from the corresponding article content in the Kopiwiki plain text extraction (Pataki et al., 2012). Terms include three million token 1-3 grams from Mikolov et al. (2013b), with the top 40 by document frequency as stop words. Candidate entities are scored using cosine distance between a mention context \vec{t}_m and the entity model \vec{t}_e :

$$f_{bow}(m, e) = 1 - \cos(\vec{t}_m, \vec{t}_e) = 1 - \frac{\vec{t}_m \cdot \vec{t}_e}{\|\vec{t}_m\| \|\vec{t}_e\|}$$

On the CoNLL development data, BOW context derived from Wikipedia article text achieves 50.6 $p@1$. We also build entity models from their mention contexts, i.e., the combined text surrounding all incoming links. We project mentions into Kopiwiki article text, which yields more contexts than actual Wikipedia links. For an article a ,

we tag as mentions all aliases of entities linked to from a . We use aliases from YAGO *means* relations (see Section 3.4.1). To ensure high precision, we only use aliases that are unambiguous with respect to the outlink set, have a length of at least two characters, include at least one upper-case character, and are not a member of the NLTK stop list. This is a noisy process, but gives us a pivot to assess whether differences observed later between Wikipedia and Web link models are due the way the context is modelled or the source of the context. The term frequency for an entity e is calculated over the concatenation of all contexts for e . BOW context derived from mentions achieves 55.8 $p@1$ on the CoNLL development data, five points higher than article text.

3.5.3.0.2 DBOW context While BOW context models have been very successful, they require exact matching between terms and a large vocabulary. Distributional approaches model terms or concepts as semantic vectors (Pereira et al., 1993). Dimensionality reduction and deep learning improve generalisation and reduce vector size (Baroni et al., 2014). He et al. (2013b) report excellent performance using entity representations that optimise the similarity between mention contexts and article text in Wikipedia. However, this approach necessitates an expensive training process and significant run-time complexity. We introduce a simple distributed bag-of-words (DBOW) model that represents context as the TF-IDF-weighted average over word vectors \mathcal{V} :

$$\vec{v}_p = \frac{1}{|\mathcal{T}_p|} \sum_{t \in \mathcal{T}_p} tfidf(t, p) \cdot \vec{v}_t$$

where \mathcal{T}_p is the set of terms in passage p , and $\vec{v}_t \in \mathcal{V}$ is the learnt word vector for term t . We use existing 300-dimensional word embeddings (Mikolov et al., 2013b) and score candidates using cosine distance between mention context \vec{v}_m and the entity model \vec{v}_e :

$$f_{dbow}(m, e) = 1 - \cos(\vec{v}_m, \vec{v}_e)$$

On the CoNLL development data, DBOW context models derived from article text and mention context achieve 49.9 and 51.2 respectively.

3.6 Web link models

The models above all have direct analogues in web links to Wikipedia articles. However, web links are a comparatively noisy source. For instance, anchors are less likely to be well-formed entity mentions, e.g., in links to `Semantic Web` we observe ‘semantic markup’ and ‘Semantic Web Activity’ as anchors. A lack of curation and quality control also allows for the misdirection of links. For example, we observe links to `Apple` the fruit where the surrounding context indicates an intention to link `Apple Inc` instead. It is an open question whether link-derived models are effective in disambiguation.

Below, we describe how models are instantiated using link data. We leverage the Wikilinks corpus of 9 million web pages containing a total of 34 million links to 1.7 million Wikipedia pages (Singh et al., 2012). This includes links to English Wikipedia pages that pass the following tests: (1) the page must not have >70% of sentences in common with a Wikipedia article; (2) the link must not be inside a table, near an image, or in obvious boilerplate material; (3) at least one token in the anchor text must match a token in the Wikipedia title; and (4) the anchor text must match a known alias from Wikipedia. The corpus provides the web page URL, the link anchor, and local textual content around each link. Refer back to Table 3.2 for $p@1$ results for individual Web link components on the development data.

3.6.1 Entity prior

To instantiate f_{prior} , we build a page-entity link graph from Wikilinks. Where pages and entities are the same in the Wikipedia graph, here we have an unweighted bipartite graph of links from web pages to Wikipedia articles (see Figure 3.3). On the CoNLL development data, the link-derived entity prior achieves 63.0 $p@1$. Table 3.3 characterises the two graphs. Note that the high entity count for Wikipedia here includes red links to articles that do not exist. The actual number of entities used in the Wikipedia model is 4.4 million. Nevertheless, while the two graphs have a similar number of pages that

	Wikipedia	Web links
Pages	8.7m	9.0m
Entities	8.9m	1.7m
Pairs	100.3m	31.2m

Table 3.3: Comparison of page-entity link graphs from Wikipedia and Wikilinks (in millions). These graphs are the basis for entity prior features (Sections 3.5.1, 3.6.1).

contain links, Wikipedia includes three times as many link pairs to 2.5 times as many entities. Furthermore, entities average 11.5 incoming links in the Wikipedia graph, compared to 3.5 in the Wikilinks graph. Nevertheless, the individual performance of the Web link prior is only 5.4 points shy of the corresponding Wikipedia prior.

Relative frequencies in Wikipedia and Wikilinks are similar, especially for entities that show up in the evaluation data. We observe a moderate correlation between entity priors from Wikipedia and Wikilinks ($\rho = 0.51$, $p < 0.01$), and a strong correlation across the subset of entities that occur in the development data ($\rho = 0.74$, $p < 0.01$).

3.6.2 Name probability

To instantiate f_{name} , we build a name-entity graph from Wikilinks. The structure is the same as the corresponding model from Wikipedia, both are bipartite graphs with cooccurrence frequencies on edges. However, names here are sourced from link anchors in web pages rather than Wikipedia articles. For comparability with the Wikipedia model, we ignore links to entities that occur fewer than five times. We observed no improvement using all links in development experiments. On the CoNLL development data, link-derived name probability achieves 58.4 $p@1$, more than ten points shy of the Wikipedia-derived name probability. Table 3.4 helps to explain this difference. Wikilinks has twice as many names linking to the same number of entities, resulting in more ambiguity and sparser models.

	Wikipedia	Web links
Names	1.4m	3.1m
Entities	1.5m	1.7m

Table 3.4: Comparison of name-entity link graphs from Wikipedia and Wikilinks (in millions). These graphs are the basis for name probability features (Sections 3.5.2, 3.6.2).

3.6.3 Textual context

To instantiate f_{bow} and f_{dbow} , we follow the same methodology used for Wikipedia mention contexts. The term frequency for an entity e is calculated over the concatenation of mention contexts for e . Document frequency is also calculated across aggregated entity contexts. Mention contexts include all text included in the Wikilinks data, a window of 46 tokens on average centred on the link anchor. Section 3.5.3 showed that Wikipedia mention contexts give better individual performance than Wikipedia article texts. Web link mentions result in even better performance. On the CoNLL development data, BOW context achieves 62.2 $p@1$, ten points higher than commonly used Wikipedia article model and seven points higher than the analogous Wikipedia mention model. DBOW context achieves 54.0 $p@1$, 2.8 points higher than the Wikipedia mention model.

Table 3.5 compares Wikipedia and Wikilinks coverage of entities from the CoNLL development set. The first column indicates the source of textual context model. The second column ($|\mathcal{E}|$) contains the number of unique entities that have usable context. Note that the entity universe we consider here is all article pages in English Wikipedia (4,418,901 total from the December 2013 Kopiwiki data set). The third and fourth columns correspond to coverage of entities ($Cov_{\mathcal{E}}$) and mentions ($Cov_{\mathcal{M}}$) from the CoNLL data set. Mention coverage exceeds entity coverage, highlighting the relationship with prevalence in newswire. The last column contains $p@1$ for the subset of mentions in CoNLL for which the correct resolution is jointly covered by both

	$ \mathcal{E} $	$Cov_{\mathcal{E}}$	$Cov_{\mathcal{M}}$	Joint
Articles	4,418,901	100	100	51.1
Mentions	954,698	77	89	58.3
Web links	1,704,703	82	92	64.1

Table 3.5: Coverage of textual context models for each source over entities (\mathcal{E}) and mentions (\mathcal{M}).

	$\bar{t}_{\mathcal{E}}$	$\bar{t}_{\mathcal{M}}$
Articles	438	438
Mentions	1653	50
Web links	922	46

Table 3.6: Mean in-vocab tokens per entity ($\bar{t}_{\mathcal{E}}$) and tokens per mention ($\bar{t}_{\mathcal{M}}$) for each textual context model.

Wikipedia articles and web links. This isolates context source, demonstrating that link contexts outperform article text.

Table 3.6 compares context size in Wikilinks to Wikipedia. The second column ($\bar{t}_{\mathcal{E}}$) contains the mean number of tokens per covered entity. The third column ($\bar{t}_{\mathcal{M}}$) contains the mean number of tokens per mention. WikilinksBOW models are approximately twice the size of Wikipedia article models and half the size of Wikipedia mention models. This helps to explain why individual mention and link models outperform individual article models.

3.7 Learning to rank

To perform disambiguation, we first extract a set of real-valued features for each candidate entity e given a training set of mentions M . Features values are standardised to have zero mean and unit variance. Parameters of the training distribution are saved for consistent standardisation of test data.

We train a Support Vector Machine (SVM) classifier to perform pairwise ranking (Joachims, 2002). For each mention in the training set, we derive training instances by comparing the feature vector of the gold link (\vec{f}_g) with each non-gold candidate (\vec{f}_c):

$$(x_i, y_i) = \begin{cases} (\vec{f}_g - \vec{f}_c, +) & \text{if } i \text{ is odd} \\ (\vec{f}_c - \vec{f}_g, -) & \text{otherwise} \end{cases}$$

For example, given a mention span of "Tesla" we may generate candidates for both the Nikola Tesla and Tesla Motors entities. We then compute feature vectors for each entity and compute the difference based on the gold standard entity assignment for each mention. As mention spans may produce a variable number of candidates, we selectively limit the number of instances per mention to the top-ten non-gold candidates by sum of absolute feature values:

$$activation(c) = \sum_{i=1}^{|\vec{f}_c|} |\vec{f}_{c,i}|.$$

In development experiments, this outperformed random selection and difference in activation. Class assignment is alternated to balance the training set.

To capture non-linear feature relationships we incorporate a degree-2 polynomial kernel via explicit feature mapping (Chang et al., 2010). Regularisation parameters are selected via grid search over the development set. Our final model utilises an L1 loss function, L2 weight penalty and SVM penalty parameter $C \approx 0.03$.

3.7.1 Feature selection

Sections 3.5 and 3.6 describe a total of ten model components, six from Wikipedia and four from Wikilinks. We select the optimal combination through exhaustive search. Figure 3.4 includes individual and cumulative results on the CoNLL development data. The article, mention and web link models each attain their best performance with all component features (entity, name, BOW, and DBOW): 84.7, 81.1, and 75.0 respectively. Adding mention context features doesn't improve the more conventional Wikipedia article model. Combining all features gives 87.7, while the optimal configuration

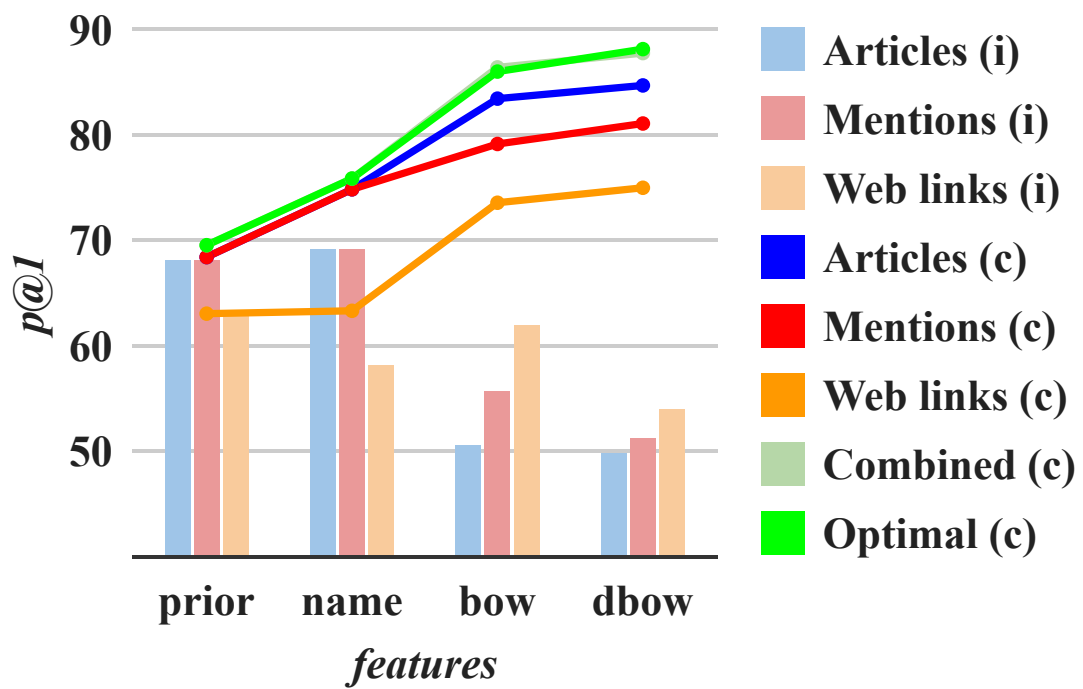


Figure 3.4: Individual (i) and cumulative (c) results for basic features on the CoNLL development data. Combined includes all features while Optimal includes the best subset. Optimal tracks Combined closely, but is just higher.

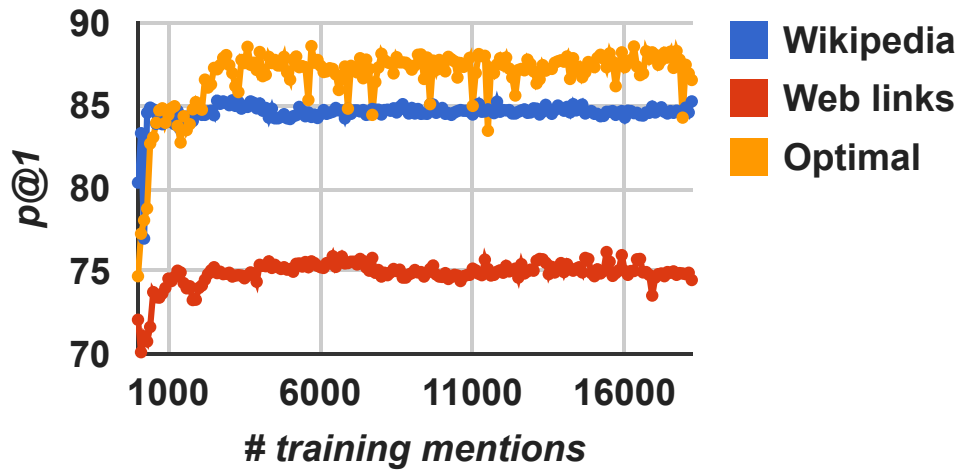


Figure 3.5: SVM learning curves for best configurations.

achieves 88.1 without Wikipedia mention contexts. In the remaining experiments, optimal refers to Wikipedia article plus web link features and Wikipedia refers to article features alone.

3.7.2 Effect of training data size

Figure 3.5 compares learning curves for each model on CoNLL development data. The x-axis corresponds to $p@1$ scores and the y-axis corresponds to the number of (randomly selected) mentions used in training. All models stabilise early, suggesting 6,000 annotated mentions are sufficient for the SVM to learn feature weights. Possibly due to higher quality and consistency of features, the Wikipedia model stabilises earlier, before 1,000 annotated mentions.

3.7.3 Ablation analysis

Figure 3.6 contains an ablation analysis for Wikipedia and Web link features, as well as the optimal overall combination of both. Here we investigate the performance of distinct model variants each trained by omitting one feature in turn. The most striking effect is due to the popularity components. Removing entity prior features reduces

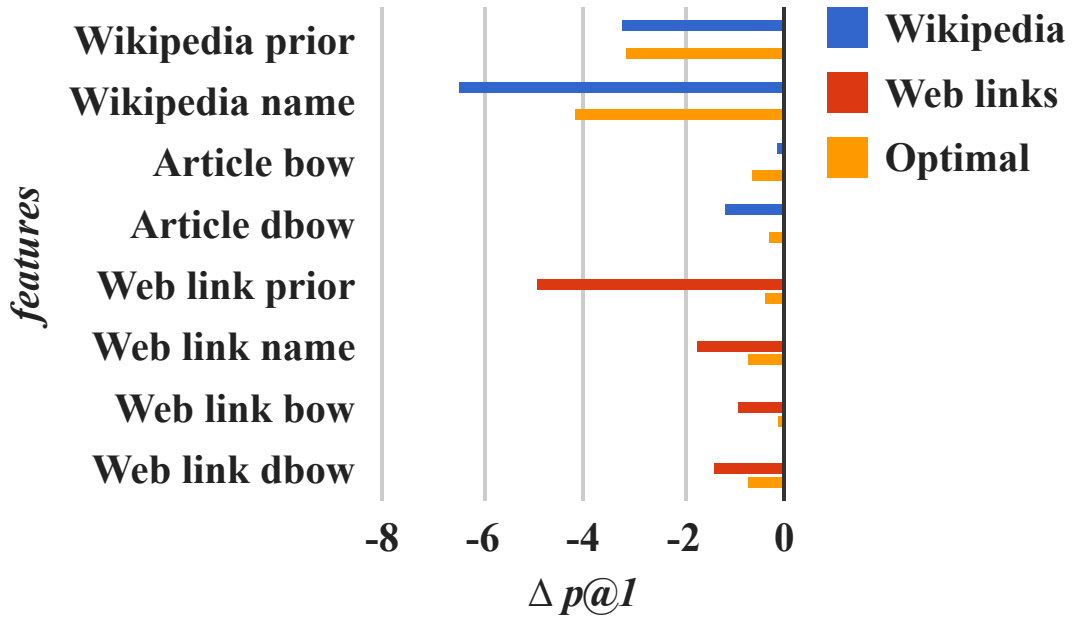


Figure 3.6: Ablation analysis of best configurations.

$p@1$ by 3.2 for Wikipedia and 5.0 for Web link. Removing name probability reduces $p@1$ by 6.5 for Wikipedia and 1.8 for Web link. In the overall model, the Wikipedia popularity components have a much larger impact (prior: -3.2, name: -4.2) than the Web link popularity components (prior: -0.4, name: -0.8). These results show the impact of noisy web links, which appears to be worse for name probability modelling. For context, removing DBOW features have a larger impact than BOW for both Wikipedia (BOW: -0.2, DBOW: -1.3) and Web link (BOW: -0.9, DBOW: -1.4). All individual context features have a small impact on the overall model despite redundancy.

3.8 Adding coherence

The model combinations above provide a strong, scalable baseline based on popularity and entity context. Another approach to context leverages the Wikipedia link graph to explicitly model the coherence among possible resolutions. Here, systems define some measure of entity-entity relatedness and maximise the coherence of entity assignments across the query document as a whole. This can be done using global methods over

the entity link graph (Hoffart et al., 2011), but these have high runtime complexity. We employ a simple approach based on conditional probabilities:

$$p_{coh}(a|b) = \frac{|\mathcal{I}_a \cap \mathcal{I}_b|}{|\mathcal{I}_b|}$$

where \mathcal{I}_e is the set of documents that link to entity e . The candidate-level feature is the average:

$$f_{cond}(e) = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \log p_{coh}(e|c)$$

where \mathcal{C} is the set of context entities for candidate entity e . For Wikipedia and Web link coherence, \mathcal{I}_e models are derived respectively from the set of other articles that link to e and from the set of web pages that link to e . Given the same initial ranking from the optimal base model, Wikipedia and Web link coherence models alone achieve 84.7 and 76.6.

3.8.1 A two-stage classifier

To incorporate coherence, we use a two-stage classifier. First, we obtain an initial candidate ranking for each mention using the basic model described in Section 3.7 above, and populate \mathcal{C} from the top-one candidate for each unique context name. A second classifier incorporates all features, including basic components and coherence. Given the same initial ranking, adding coherence improves individual Wikipedia and Web link models 4.5 and 6.4 points to 89.2 and 81.4 $p@1$ on the CoNLL development data. These results suggests that coherence is a powerful feature to overcome low scores in the basic Web link model. But, coherence only improves the optimal combination of basic Wikipedia and web link features by 1.1 point to 89.2. This suggests our formulation of coherence does not contribute much on top of strong set of basic context models.

3.9 Final experiments

We report final experiments on the held-out CoNLL and TAC 2010 test sets. As described in Section 3.4 above, we report $p@1$ for CoNLL following Hoffart et al. (2011) and A_{kb}

	(a) CoNLL		(b) TAC 10	
	Pop	Ctx	Pop	Ctx
Wikipedia	73.9	53.3	72.6	65.0
Web links	62.5	60.8	73.3	75.3

Table 3.7: Web link components vs. Wikipedia.

for TAC following He et al. (2013b). We use a reference implementation to compute evaluation measures and pairwise significance (Hachey et al., 2014). We bold the superior configuration for each column only if the difference is significant ($p < 0.05$).

3.9.1 Results

3.9.1.0.1 Can link components replace KB components? Table 3.7 compares performance of basic model components. The popularity (Pop) column contains results using just entity prior and name probability features. The context (Ctx) column contains results using just BOW and DBOW features. Results follow trends observed in development experiments. Specifically, Wikipedia popularity models are better, but web link context models are better. Interestingly, web link popularity is significantly indistinguishable from Wikipedia popularity on TAC 10 data. This may be attributed to the fact that TAC selectively samples difficult mentions.

3.9.1.0.2 Can links replace a curated KB? Table 3.8 compares performance of the Wikipedia and Web link systems using the basic feature set alone and with coherence. Wikipedia models generally perform better. However, the Web link configurations perform at 93.1, 95.1, 99.9, and 100% of the Wikipedia linker – 97% on average. This suggests that a link data set can replace a curated KB, with only a small impact on accuracy. Results also show that adding coherence improves performance in all cases.

3.9.1.0.3 Do links complement article text? Table 3.9 compares a standard Wikipedia-only model to a model that also includes features derived from Web link data.

	(a) CoNLL		(b) TAC 10	
	Base	+Coh	Base	+Coh
Wikipedia	82.7	84.9	78.6	80.2
Web links	77.0	80.7	78.5	80.2

Table 3.8: Web link combinations vs. Wikipedia.

	(a) CoNLL		(b) TAC 10	
	Base	+Coh	Base	+Coh
Wikipedia	82.7	84.9	78.6	80.2
+ Web links	86.1	88.7	79.6	80.7

Table 3.9: Web links complement Wikipedia.

Adding Web link data has a strong impact on CoNLL, improving both configurations by approximately 4 points. We observe less impact on TAC. Nevertheless, the large improvements on CoNLL provide good evidence for complementarity and recommend using both feature sets when available.

3.9.1.0.4 The state of the art Finally, Table 3.10 compares our Wikipedia and Web link combinations to state-of-the-art numbers from the literature. First, we note that adding coherence to our base model results in a significant improvement on CoNLL test data, but not on TAC 2010. For comparison the literature, we report 95% confidence intervals. If a confidence bar overlaps a reported number, the difference can not be assumed significant at $p < 0.05$. Results on TAC 10 are competitive with He et al. (2013b) 81.0. On the CoNLL data, our best system achieves 88.7 $p@1$ —a new state of the art. Furthermore, the best base model is competitive with previous art that uses complex collective approaches to coherence.

	DEV	CoNLL	TAC 10
Base model	87.7	86.1	79.6
- 95% CI	[85.3, 90.0]	[83.1, 88.8]	[77.1, 82.1]
Base+Coh	89.4	88.7	80.7
- 95% CI	[87.3, 91.2]	[86.2, 90.9]	[78.2, 83.1]
Hoffart	79.3	82.5	—
Houlsby	79.7	84.9	—
He	—	85.6	81.0
Alhelbawy	—	87.6	—

Table 3.10: Comparison to the disambiguation literature.

3.10 Discussion

We set out to determine whether links from external resources can replace a clean, curated KB. Wikipedia is an incredible resource that has advanced our understanding of and capabilities for identifying and resolving entity mentions. However, it covers only a small fraction of all entities. Applications that require other entities must therefore extend Wikipedia or use alternative KBs. We explore a setting where a custom KB is required, but it is possible to harvest external documents with links into the custom KB. Overall, results are promising for using links in a knowledge-poor setting. The link-derived system performs nearly as well as the rich-KB system on both of our held-out data sets.

Web link combinations perform at 97% of Wikipedia combinations on average. However, creating a KB as rich as Wikipedia represents an estimated 100 million hours of human effort (Shirky, 2010). We do not have a comparable estimate for the Web link data. However, it is created as byproduct of publishing activities and the labour pool is external. Considering this and the additional noise in web data, it is remarkable that the Web link models do so well with respect to the Wikipedia models.

We also present detailed experiments comparing popularity, context, and coherence components across settings. Here, results are even more surprising. As expected, Web link popularity and coherence models trail Wikipedia models. However, Web link context models outperform Wikipedia context models by 7 to 10 points.

We add the Web link components into the Wikipedia system to achieve a result of 88.7 on the CoNLL data set, the best reported result at the time. Still, our results suggest that coherence modelling does not require complex global graph algorithms. Our simple approach improves performance over the basic model by one to three points. On the other hand, our basic system without coherence modelling approaches state-of-the-art performance on its own. This suggests that additional popularity and context features from web links can replace coherence where efficiency is a concern. In subsequent work, deep neural network models which jointly embed entities and context terms (Yamada et al., 2016) and selectively attend context representations (Ganea and Hofmann, 2017) have significantly improved upon our results on this benchmark.

We believe these results have a number of implications for management of entity KBs. First, they motivate concerted efforts to link content to KBs since links lead to substantial accuracy improvements over a conventional model based on rich KB data alone. Second, it informs allocation of editorial resources between interlinking data sets and curating KBs. Since models built from link data alone approach state-of-the-art performance, curating links is a reasonable alternative to curating a KB. This is especially true if link curation is cheaper or if links can be created as a byproduct of other content authorship and management activities.

Finally, where KB data is currently proprietary, results here motivate openly publishing KB entities and encouraging their use as a disambiguation endpoint for public content. In addition to providing pathways to paid content, incoming links provide a simple means to harvest rich metadata from external content and this can be used to build high-quality resolution systems.

A key avenue for future work is to evaluate how well our disambiguation approach fits into the broader entity linking pipeline. End-to-end entity linking performance is highly dependant on a pipeline of components including NER (Hachey et al., 2013; Ling et al., 2015). We expect web links to provide similar benefits to NER systems. Linked mentions are a potential source of NER training data (Nothman et al., 2008) and link anchors provide a source for entity name gazetteers — a crucial component of high-performance NER systems (Ratinov and Roth, 2009).

3.11 Summary

Despite widespread use in entity linking, Wikipedia is clearly not the only source of entity information available on the web. We demonstrate the potential for web links to both complement and completely replace Wikipedia derived data in entity linking. This suggests that, given sufficient incoming links, any knowledge base may be used for entity linking. In subsequent chapters we develop this idea, exploring how KB-like structures emerge as a feature of the web as a whole.

4 Web Knowledge Base Discovery

Getting information off the Internet is
like taking a drink from a fire
hydrant.

Mitch Kapor

Recognition and disambiguation of named entities in text is a knowledge-intensive task. To fill this knowledge gap, systems typically leverage the resources of a structured knowledge base in entity disambiguation. These resources provide context for entity modelling, but impose an upper bound on recall given their domain of entity coverage. While KBs like Wikipedia continue to expand in both size and scope, this growth is limited by the availability of dedicated human editors who create and maintain content. In Chapter 3, we described how web links can provide an alternative knowledge source for entity disambiguation with Wikipedia. This approach increases the depth of available entity knowledge; improving NED performance for Wikipedia entities — but does not improve the breadth of entity coverage beyond Wikipedia’s bounds.

In this chapter we extend the idea of inlink driven entity disambiguation to the broader domain of entity knowledge available on the web. Within this setting, Wikipedia is just one of many aggregation points for entity references. These resources present an opportunity for collecting a far broader set of human annotated entity mentions than any single dedicated KB can provide. However, to actually exploit these resources, we must first infer their existence on the web. We explore a data driven approach. Given a corpus of linked documents on the web, we attempt to infer a set of

URL patterns which reliably disambiguate named entity mentions. We refer to these URLs and the pages they target as **entity endpoints**. While all links on the web clearly do not represent disambiguation endpoints, patterns which reliably produce entity mentions can provide evidence for the existence of KB-like structure. In this chapter we develop and evaluate a systems which automate the discovery of these resources.

Contributions include: (1) a formalization of the Knowledge Base Discovery (KBD) task and investigation of KB-like structure on the web; (2) exploration of an classification framework for KBD and implementation of a KBD system; (3) detailed development experiments and crowd-sourced endpoint annotations for evaluating KBD systems. Experiments from this chapter were first described in Chisholm et al. (2016b). Code and evaluation data are available under an open source licence at: github.com/andychisholm/web-kb.

4.1 Introduction

Linking systems typically draw upon a single store of semantic knowledge in entity disambiguation. However, this limits their scope of entity coverage. Wide domain KBs like Wikipedia cover a diverse set of entities, but constrain coverage to only notable entities. On the other hand, narrow domain resources like IMDb¹ or MusicBrainz² provide deep coverage down the tail of entity notability at the expense of breadth. While it is sometimes possible to merge resources across structured KBs for a given application, reconciling distinct entity sets is often difficult. Even when explicit linking between KBs is present (e.g. Wikidata and Freebase), merging knowledge across distinct KB schema can be problematic.

As an alternative to merging structured data, we relax the definition of a KB to include any URL on the web which reliably disambiguates inbound web links. Under this definition, we are able to leverage resources which both work as a KB by design

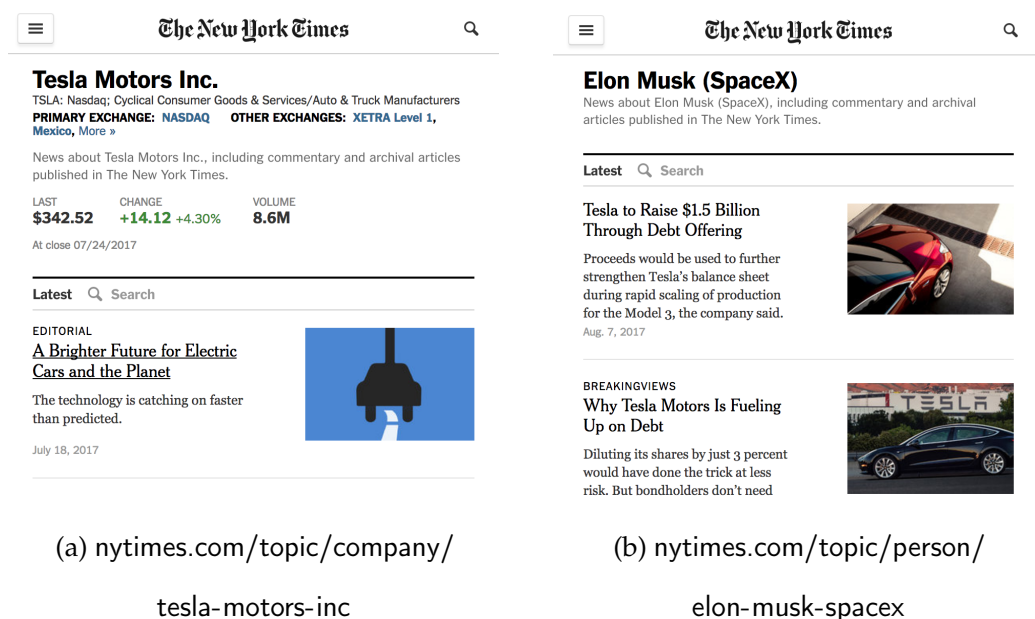
¹<http://www.imdb.com>

²<https://musicbrainz.org>

(e.g. Wikipedia articles) and those which do so implicitly by disambiguating inbound mentions. Take for example the following snippet:

For almost three months, [Tesla](#) reigned as the most valuable automaker in the nation, ahead of both General Motors and Ford Motor, thanks to a remarkable run-up in its stock this year. It seemed its chief executive, [Elon Musk](#), could do no wrong.

Figure 4.1: Annotated links to Tesla Motors and Elon Musk on nytimes.com



In this passage, mentions of both “Tesla Motors” and “Elon Musk” have been annotated with web links by the author. Crucially, these links both target URLs under the `nytimes.com/topic/*` endpoint. The motivation for content publishers is clear — links provide an aggregation point for news stories about an entity and help drive clicks to related content and retain user attention. However, this style of annotation is equally useful as a mechanism for recognizing and disambiguating ambiguous named entity mentions. Given knowledge that links targeting URLs under `nytimes.com/topic/*` represent entities, we are able to leverage the content and context of inlinks from both `nytimes.com` articles and the rest of the web in the same way we leverage inlinks to a dedicated KB like `wikipedia.org`.

This style of systematic entity indexing is common on the web³. It is a characteristic feature of social sources (e.g. `twitter.com/*`), news aggregation endpoints (e.g. `bloomberg.com/quote/*`) and organization directories (e.g. `sydney.edu.au/engineering/people/*`). These resources present a valuable and largely untapped source of entity information, both in the content they host and semantic resources that may be extracted from aggregated inbound links. Moreover, they have the potential to index many entities which don't otherwise warrant entry in a major KB. For every KB pattern we uncover on the web (e.g. `twitter.com/*`) we may infer both the existence of new entities through derived endpoint targets (e.g. `twitter.com/Tesla_Motors`) and recover textual mentions of these entities through inbound web links.

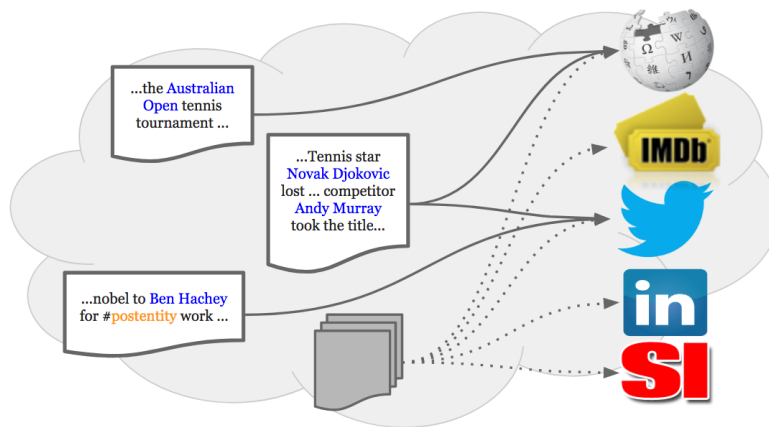


Figure 4.3: Links targeting entity endpoints across multiple web KBs.

In this section, we propose a method for automatically discovering web KBs given a corpus of linked documents from the web. Our experiments suggest that a weakly-supervised KBD model can classify candidate endpoints with a precision of 71.2% in a crowd-sourced post-hoc evaluation. We also answer the question of whether these endpoints uncover entities outside the bounds of a major KB, finding that 20% of inferred entity targets reference novel entities outside Wikipedia.

³<https://cloudplatform.googleblog.com/2017/10/API-design-choosing-between-names-and-identifiers-in-URLs.html>

4.2 Related work

Entity linking and wikification have typically relied on Wikipedia (Cucerzan, 2007; Milne and Witten, 2008) or a subset (McNamee et al., 2009), or a larger structured resource such as Freebase (Zheng et al., 2012b). Entries in the KB provide a point against which mentions that refer to that entity are clustered. In addition to this, the KBs provide extra information for an entity such as facts, text and other media. Hachenberg and Gottron (2012) address the reverse task of identifying *good links* that correspond to specific KB entities by searching for the entity name in a web search engine and refining the results.

Other tasks cluster mentions of the same entity, but without reference to a central KB, namely Cross Document Coreference (Bagga and Baldwin, 1998; Singh et al., 2011) and Web Person Search (Artiles et al., 2007). These tasks can be more challenging, as we are unable to exploit priors inferred from the KB or leverage information about an entity for clustering. While KBs and a set of coreference clusters are quite different, they both act as *aggregation* points for mentions of their respective entities.

Mining the content and structure of pages to discover new entities is another important task. There is also substantial work in trying to identify instances of entity classes from text, exploiting language (Hearst, 1992) document structure (Wang and Cohen, 2007; Bing et al., 2016) and site structure (Yang et al., 2010). Clustering NIL entities (those that cannot be linked to the KB) has been a focus of the Text Analysis Conference (TAC) Knowledge Base Population shared tasks from 2011 (Ji et al., 2011). This work is important for growing KBs to include more entities about which we know less – i.e. the long tail.

We examine whether we can successfully extract informal web KBs by exploiting the structure of individual URLs and the structure of the sites they describe. Like traditional linking URLs, they identify reference points against which mentions can be linked, but lack the information commonly expected in URLs.

4.3 Web entity endpoints

In this section we describe common patterns on the web producing endpoints for entity disambiguation. These endpoints vary widely in terms of the type of content they host, the kinds of inlinks they accumulate and the domain of entities they cover.

4.3.1 Online encyclopedia

The first and perhaps most prominent form of disambiguation endpoint we observe on the web are online encyclopedia. While we have already considered applications of inlinks into Wikipedia, many other similar resources index information about entities on the web. Crowd-sourced wikis with deep coverage of specific verticals are common. For example, fishbase.com for fish species, memory-alpha.wikia.com for the fictional Star-Trek universe or wiki.teamliquid.net for E-sports athletes and teams. Other sites summarize entities (e.g. biography.com) or aggregate structured facts and statistics — especially those covering sporting teams and players (e.g. si.com, sports.yahoo.com). These resources are a rich source of entity information, both in terms of the inbound web links they accumulate and the content of the endpoint page itself.

4.3.2 Web news

Some publishers maintain topic pages that aggregate structured and unstructured content on entities, e.g., nytimes.com/topic/person/barack-obama. These provide a landing page for search engine optimisation and enable some semantic analytics (e.g. “Do users click more on people than organisations?”). They also provide a link target to contextualise mentions in news articles and help prevent navigation away from the site. Notably, these pages may not include a description of the entity, merely aggregated content. In cases where endpoints aggregate loosely defined tags, the specific target must be taken into account when classifying a URL. For example, breitbart.com/tag/donald-trump may represent an entity but breitbart.com/tag/big-govenment does not.

4.3.3 Social networks

Social sites are rich source of entity information, e.g., [linkedin.com/in/barackobama](https://www.linkedin.com/in/barackobama) . In particular, they offer a view down the long-tail of entities addressable on the web. While Wikipedia indexes on the order of 1-2M notable person entities, Facebook claims to have surpassed 2B active users in 2017. Each of these users entail an addressable profile page on the web, though access and privacy controls may restrict the content of the page itself. Our analysis identifies some of these endpoints.

One challenge for social profile links in particular is a tendency towards anchors that are not mentions of the target entity, e.g., "Find me on [Twitter](twitter.com/john-smith)." This pattern in linking violates our the assumption that all named entity inlinks to an endpoint represent mentions of that entity. Despite this, the broader document context surrounding the link may still be informative in disambiguation. For preliminary experiments described in this chapter we simply ignore these patterns. We address this issue in part when revisiting KBD as part of experiments in Chapter 5 (see: 5.7.1).

4.3.4 Organisation directories

Universities, law firms and other professional organizations often maintain directories of employee profiles, e.g., gtlaw.com/People/Matthew-Galati . These collect fewer inlinks than news site topic pages and social profile pages. They are nevertheless a promising source of information for entities that don't meet Wikipedia's notability requirements. In particular, they often present contact information, photos and descriptions of professional activity for a person.

4.4 Framework

We define an entity endpoint as any URL for which inlinks reliably identify and disambiguate named entity mentions. For example, we may observe that inlinks to

`en.wikipedia.org/wiki/Elon_Musk` are typically mentions of the entity Elon Musk. Links targeting this URL in reference to some other entity are unlikely, so we should consider this an endpoint for the entity Elon Musk. Web endpoints also yield disambiguated entity mentions. For every entity endpoint we discover, we may recover thousands of entity mentions via inlinks. While the effectiveness of Wikipedia inlinks in entity disambiguation is discussed in 3, we aim to extend this approach to leverage inlinks for a collection of automatically discovered web KBs. This process has the potential to both improve EL accuracy for well-covered entities and extend the coverage of EL systems by uncovering endpoints for previously unseen entities.

While it might be possible to manually curate a list of websites which are known to behave as entity endpoints, the web itself presents a constantly moving target. New pages are constantly being created, and updates to sites over time change the structure of existing resources. We instead propose a data driven approach to endpoint discovery. Specifying our criteria for endpoint inference and optimizing a model under this objective enables automated upkeep of disambiguation resources over time. Moreover, it allows us to uncover lesser known sites which may act like a KB in practice, even where that is not their primary intent.

4.4.1 Endpoint inference

We explore a weak supervision framework for KBD. For a web anchor span linking to a specific URL u , we wish to model the probability that it both references an entity e and is a true named entity mention m . While it may be natural to consider directly learning a model which estimates $P(e, m|u)$ given a corpus of annotated endpoint URL instances, modelling this distribution directly may be problematic. Entity endpoints make up the minority of the natural URL distribution. Even in news, a very rich source of linked entity mentions, endpoint URLs only account for 15% of links in a random sample of 200 URLs. This is problematic for cases where the structure of a candidate URL is uninformative. For example, we might reasonably estimate

that `example.com/person/john-smith` is an entity link without previously observing samples from `example.com/person/*`. It is however difficult to reliably estimate whether `example.com/e/123` represents an entity without positive samples from that domain. These factors inflate the number of annotated samples needed to train a robust model with reliable estimates over a broad set of URLs.

As an alternative to this approach, we explore a framework for automatically generating a large silver-standard dataset of annotated endpoint URL instances. In place of directly modeling $P(e, m|u)$, we instead aim to model $P(m|u)$ — the probability that a u is linked with an entity mention m .

$$\begin{aligned} P(e, m|u) &= \frac{P(e, m, u)}{P(u)} \\ &= P(e|m, u)P(m|u) \end{aligned}$$

If we take $P(e|m, u) \approx 1$ by assuming all mentions are entity references independent of their target URL, we allow for an estimation of our target distribution via a model which predicts the probability that links targeting u are a mention m .

$$P(e, m|u) \approx P(m|u)$$

In practice, we find this approximation still achieves good results. To train a this alternative model, we need only find instances where the URL anchors are entity mentions. Here we may automatically annotate a huge corpus of samples by running a NER system over unlabeled text from the web. In cases where the anchor of a URL is tagged as a named entity mention by NER we generate a positive instance for the target URL, otherwise we generate a negative instance.

4.4.2 Features

We represent endpoint URL patterns as a bag of binary features hashed to 500,000 dimensions to help manage model size. This section describes the two major categories of features used to represent instances.

URL	Features
nytimes.com/topic/person/elon-musk	person , topic , <domain>/person topic/person , person/<eid>
nytimes.com/2017/01/02/us/politics/..	NNNN , NN , us , politics <domain>/NNNN , NNNN/NN NN/NN , NN/us , us/politics

Table 4.1: Example of path features generated for sample URLs

4.4.2.1 Path Features

We tokenize endpoint patterns by splitting on forward slash characters and include path component uni-gram and bi-grams as features. To reduce sparsity, features are generated over a normalized representation of the target URL. The domain name is replaced with a special <domain> token and path terminator is replaced with <eid>. E.g. `wired.com/tag/tesla-motors` becomes `<domain>/tag/<eid>` .

We find path tokens are a good predictor of entity mentions and often generalize across KBs. For example, it is common to observe links to entity pages prefixed by terms like `/profile` or `/wiki` . Similarly, terms like `news` or date patterns `YYYY/MM/DD` in a URL can provide negative evidence. Table 4.1 shows path features generated for a set of sample URLs.

4.4.2.2 Domain Features

In many cases, patterns are not sufficient to identify a KB endpoint without prior knowledge. For example, `twitter.com` entities are only observed via a common `<domain>/<eid>` pattern. We allow the model to explicitly memorise candidate KB URLs by including as features the conjunction of domain name with each bi-gram feature. While this subset of features cannot generalise to unseen domains, we are able to achieve high precision for endpoints observed in our automatically generated seed corpus.

Endpoint Prefix	Inlinks
sfgate.com/search	330,263
blogs.reuters.com/search	131,051
twitter.com	69,022
et.indiatimes.com/topic	55,064
huffingtonpost.com/news	47,571
seekingalpha.com/symbol	45,531
facebook.com	41,678
abcnews.go.com/topics/news	37,425
linkedin.com/company	32,087
sports.yahoo.com/soccer/players	31,091

Table 4.2: Top mention-aligned URL prefixes in the seed corpus.

4.5 Dataset

For experiments described in this chapter, we utilize a proprietary corpus of 2,948,841 web news articles — HG-NEWS (Cadilhac et al., 2015). While this dataset is not publicly accessible, we will later reproduce our methodology on a larger open-access corpus of web documents for experiments described in Chapter 5.

We leverage named entity recognition to identify likely entity references in link anchors that align to predicted mentions for person, location and organisation entity types. We also map target URLs to endpoint patterns by first normalising to lower case, removing protocol (e.g., http) prefixes, port identifiers and tracking parameters (e.g. &utm_source=facebook). Table 4.2 lists the top-10 URL prefix patterns by NER-aligned inlink count. While many of sites represent entity endpoints (e.g. linkedin.com/company), many still do not (e.g. huffingtonpost.com/news).

Table 4.3 includes statistics of the full link corpus (Total) and the NER-aligned subset (Aligned). The full corpus includes a total of 14,462,659 links. 3,436,033 of these align to NER mentions, yielding 1,029,405 candidate entity endpoints across 309,182 distinct URL patterns.

	Total	Aligned
$ Mentions $	14.5	3.4
$ URIs $	5.4	1.0
$ Anchors $	4.4	0.6
$ Patterns $	1.5	0.3

Table 4.3: Statistics of the corpus in millions. The first column includes all corpus links.

The second column includes links whose anchor text aligns to an NER span.

4.6 Model

We estimate $P(m|u)$ via logistic regression using a sample of (u, m) pairs that act as a silver standard. We consider all URL patterns with ten or more inlinks as possible training instances. We treat a URL pattern as a positive instance if a majority of inlinks from our corpus are aligned to mentions. If not, we treat it as a negative instance. To estimate performance on unseen URL patterns, we group instances by domain name before partitioning into training and development test sets. This produces a silver standard training set of 100,852 instances (10% positive), and a development test set of 10,404 (12% positive). Before training, we subsample positive instances in the training data to equal the number of negative instances.

4.6.1 Development experiments

We select a threshold on held out instances from our development split. Figure 4.4 shows the precision-recall trade-off across possible threshold values. We observe a slight plateauing of recall between 0.52 and 0.47 at threshold values in the range $[0.725, 0.875]$. In this same range, precision goes up from 0.70 to 0.94. We select a threshold of $P(m|u) \geq 0.825$ here as this maximises F-score at 0.64 and is in the middle of the threshold range.

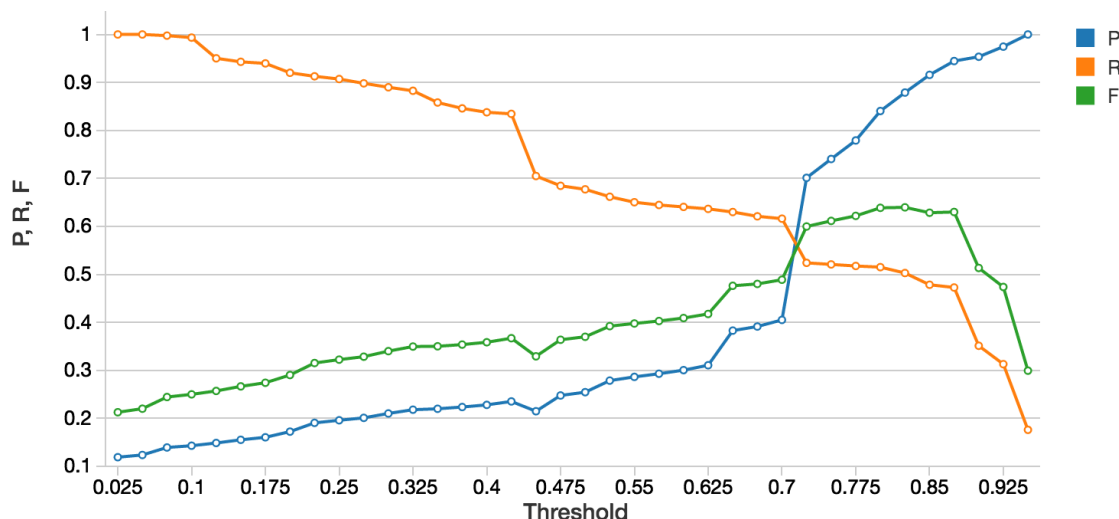


Figure 4.4: Precision-recall trade-off across thresholds.

4.6.2 Analysis

Table 4.4 shows endpoint probability estimates from our model over an illustrative selection of notable web endpoints. The model assigns strong estimates to URLs with an informative path features (e.g. terms like person, player or news). Interestingly, our model assigns low estimates for patterns like `facebook.com/<eid>`. Inspecting inlinks for these targets, we find many anchors do not constitute well formed entity mentions, e.g. "Check out our [Facebook page]" or "Visit the [Official Site]". A similar problem exists for endpoints like `wikipedia.org` which have both entity and non-named entity targets, e.g. "a [self-governed](`en.wikipedia.org/wiki/Self-governance`) territory". Non-named entity targets under a shared endpoint pattern generate negative instances and thus pull down the endpoint probability estimate under our model. We attempt to address this issue when revisiting KBD as part of our experiments in Chapter 5 (see: 5.5).

Table 4.5 shows sample URL patterns predicted by the model alongside the number of matching entity URLs in the seed corpus. Encouragingly, apart from general news, we see two of the endpoint categories from Section 4.3: domain-specific news topic

URL	Normalized Endpoint Pattern	$P(m u)$
nytimes.com/topic/person/john-smith	nytimes.com/topic/person/<eid>	0.9625
si.com/college-football/player/john-smith	si.com/college-football/player/<eid>	0.9285
linkedin.com/in/johnsmith	linkedin.com/in/<eid>	0.9281
variety.com/t/phoenix/	variety.com/t/<eid>	0.8994
linkedin.com/company/johnsmithco	linkedin.com/company/<eid>	0.8874
twitter.com/johnsmith	twitter.com/<eid>	0.8256
en.wikipedia.org/w/index.php?id=123	en.wikipedia.org/w/index.php/<eid>	0.6920
en.wikipedia.org/wiki/johnsmith	en.wikipedia.org/wiki/<eid>	0.5277
facebook.com/johnsmith	facebook.com/<eid>	0.4530
twitter.com/johnsmith/status/123	twitter.com/johnsmith/status/<eid>	0.3091
nytimes.com/2016/03/23/world/story.htm	nytimes.com/NNNN/NN/NN/world/<eid>	0.0482

Table 4.4: Model estimates for notable endpoints

Endpoint	Entities
linkedin.com/in	3,246
variety.com/t	2,871
data.cnbc.com/quotes	2,958
si.com/nfl/player	1,426
ign.com/stars	933
cyclingnews.com/riders	899
gtlaw.com/people	257

Table 4.5: Sample of predicted URL patterns and entity counts.

pages from Sports Illustrated and Cycling News, and professional profile pages like LinkedIn and legal web sites, which can inform disambiguation models for long-tail entities.

4.7 Evaluation

To evaluate how well our model for $P(m|u)$ estimates $P(e, m|u)$, we construct a corpus of human-annotated endpoint URLs. In constructing this corpus, we also all seek to investigate questions of endpoint redundancy. Specifically, do we find multiple endpoints describing the same entity, and what portion of web KB entities are already covered by a major KB such as Wikipedia.

We design a crowd task to collect pairwise identity judgments within clusters of candidate coreference pairs. To build clusters, we retrain our model over combined silver standard data (train + test) and use it to collect endpoints from the complete seed corpus with classification confidence above our threshold. While it would be possible to randomly sample URLs, this would give us a highly imbalanced set with very few positive instances of coreference between endpoint pairs. We instead focus on entities which are connected via a common anchor, and are thus far more likely to be coreferent than not.

We construct a graph of anchors and URLs vertices and add edges between nodes whenever we observe a distinct link-anchor pair in our dataset. We then sample pairs by first sampling a seed URL, then randomly walk up-to four steps through the anchor-URL graph to another URL node. To the extent that anchors reliably encode the name of linked entities, we expect this method to return a mix of endpoints for both aliases of an entity name and ambiguous entities that share a name in common.

4.7.1 Crowd task

We post 500 URL pairs to Crowdflower⁴ and ask three workers to judge whether each endpoint is an entity page. We also ask whether they refer to the same underlying entity. The task interface is show in Figure 4.5. In the task introduction we describe what does and does not constitute an entity page and provide several sample endpoints.

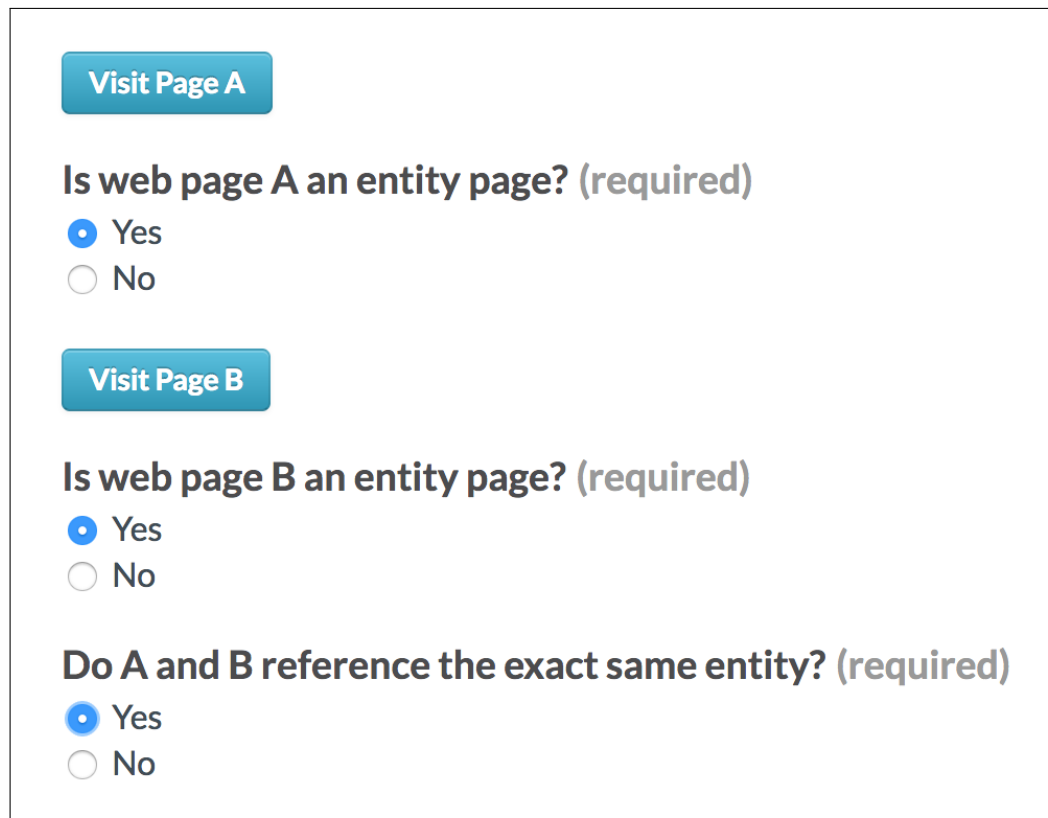
⁴<http://www.crowdflower.com>

We constrain our task to use the highest reputation tier for workers and configure 11 test questions to help filter unreliable responses. We also run a small trial task to assess our description of the problem and gather feedback from annotators. Here we adjust our task description to specifically address cases of websites which do represent entities (e.g. `sydney.edu.au/about-us.html`), but do not follow the typical class-instance URL pattern of other examples.

Assessing whether or not a web page represents an entity can be a nuanced task. We find that 30% of candidate workers are dropped under test questions and a further 5% are dropped by inter-annotator heuristics throughout the task. We observe most label confusion is generated by news articles which describe events closely related to an entity. For example, almost 40% of candidates incorrectly label `www.espn.com/nba/story/_/id/13382086/roy-hibbert-looking-career-resurgence-los-angeles-lakers` as an entity page in the test set. Handling *events* in addition to *entity* pages is an interesting direction for future work as these pages often describe emerging entities and relations.

4.7.2 Results

We collect a total of 1,500 trusted judgments (3 per question) at a cost of \$38 USD. After labeling instances by majority vote, we observe that 71.2% of candidate endpoints are confirmed as entities. Of the 277 pairs that include two true endpoints, 70.8% are judged as coreferent. Finally, we sample 100 validated endpoints and manually search for a corresponding Wikipedia article. We find that 20% of endpoints represent entities that are not already in Wikipedia. This suggests that our approach does discover useful knowledge further down the tail of notability.



Visit Page A

Is web page A an entity page? (required)

☒ Yes
☐ No

Visit Page B

Is web page B an entity page? (required)

☒ Yes
☐ No

Do A and B reference the exact same entity? (required)

☒ Yes
☐ No

Figure 4.5: Interface for endpoint evaluation on CrowdFlower. Workers must press the "Visit Page" button and manually inspect each candidate web page. The third question addressing coreference is only displayed if a worker responds "Yes" to the preceding endpoint question for each candidate URL.

4.8 Discussion

Our model is trained to recognize endpoint URLs by predicting which URL patterns are most likely to be linked with a named entity mention in the anchor. Under this objective, we are able to automatically tag a seed corpus of 2.9 million web news articles with silver-standard NER mentions and train a model which estimates how likely a given URL is an entity endpoint.

Our crowd-sourced post-hoc evaluation suggests our model is capable of identifying endpoints with high-precision. Moreover, the acquisition of new entities is a key motivator for investigating diverse web-KBs and we find that approximately 20%

of classified endpoints reference entities outside of Wikipedia. In conjunction with the results described in Chapter 3, we believe this work motivates the curation of endpoint URL patterns as a scalable alternative to isolated mention annotations in entity knowledge tasks.

We also evaluate how often neighbours in the anchor-endpoint URL graph are coreferent. Our preliminary evaluation shows that as many as 70.8% of sampled pairs are coreferent. This confirms that there is significant redundancy in entity coverage across web KBs. Moreover, it confirms that a naive clustering of references by anchor alone is insufficient to reliably aggregate coreferent entity endpoints on the web. In subsequent work, we build on the KBD system described in this chapter to develop a shared task evaluation of systems for coreference resolution across web KB endpoints (Chisholm et al., 2016a). We consider this task and related work in detail in Chapter 5.

There are clear future directions for improving KBD. In addition to assessing the structure of an endpoint URL, we may also leverage the content of the page itself. While this increases the complexity of endpoint discovery, page content is generally decisive in resolving endpoint classifications, especially where the URL itself provides little evidence. Even without content features, we may still improve upon our KBD model. In particular we may leverage features of the terminal endpoint identifier in addition to the root URL pattern. For example, the presence of a person name John in `twitter.com/John_Smith`. We explore this extension to our base KBD model in Chapter 5.

In evaluation our constrained sampling of a relatively balanced set of positive and negative endpoint instances results in a primarily precision-oriented evaluation. We expect there will be a need to develop a significantly larger randomly sampled dataset of pages links with endpoint annotations to reliably assess KBD system recall.

Given a mechanism for discovering endpoints on the web, it is natural to consider how the distribution of web entities differs to those found in a traditional KB. In this

chapter we include only consider a brief qualitative survey of discovered entity types, leaving a more rigorous census of web entities as an interesting area for future work.

4.9 Summary

This chapter investigates the task of web knowledge base discovery. We introduce a framework for learning to automate KBD using only unlabeled data from a corpus of documents containing web links. Our findings suggest that KBD from unlabeled data alone is not only feasible, but has the potential to uncover a large number of entities which aren't otherwise indexed by a major KB. Our preliminary analysis suggests that while we may easily discover new entities through KBD, redundancy in entity converge across KB boundaries presents a distinct challenge. In the next chapter, we extend our KBD framework to a larger sample of the web and further develop the task of resolving coreference across discovered entity endpoints.

5 Cross-KB Coreference Resolution

It's not complicated, it's just a lot of it

Richard Feynman

In Chapter 4 we describe a framework for discovering URLs which represent entities on the web. For every endpoint we discover this way, we are able to aggregate textual knowledge about entities from inlinks across the web. While these resources present a potentially rich source of unstructured knowledge in downstream IE tasks, we quickly run into a problem in extending this framework to multiple web KBs — how can we consolidate references to the *same* entity across *different* KBs?

In this chapter, we explore the task of **Cross-KB Coreference Resolution** (KB-Coref). Building on our framework for KBD, we start with a corpus of unlabeled documents from the web, then proceed to extract likely entity endpoints and train a model which resolves pairwise coreference by comparing the context of inbound web links. We then develop an agglomerative clustering baseline which incrementally aggregates pairwise coreference decisions into full clusters of coreferent URLs.

Our contributions include: (1) construction of two large-scale web document collections derived from CommonCrawl data; (2) development of an inlink-driven pairwise KB-Coref resolution model; (3) annotated data for evaluation and analysis of KBD and KB-Coref baselines on CommonCrawl derived datasets;

This chapter describes preliminary work which has not previously been published under peer review. Results of a shared task we organized to benchmark long-tail pairwise KB-Coref are described in Chisholm et al. (2016a) and summarized herein

alongside related work in Section 5.2. Datasets derived from the CommonCrawl corpus including extracted document text, stand-off link annotations and endpoint probabilities are available via Amazon S3. We provide instruction for accessing the datasets, annotations, URL clusters and code under: github.com/wikilinks/sift.

5.1 Introduction

While entity endpoints are constrained by definition to uniquely identify a single entity, there may be many such endpoints for an entity across the web. In Figure 5.1 we show three different web pages representing the same underlying Tesla Motors entity.

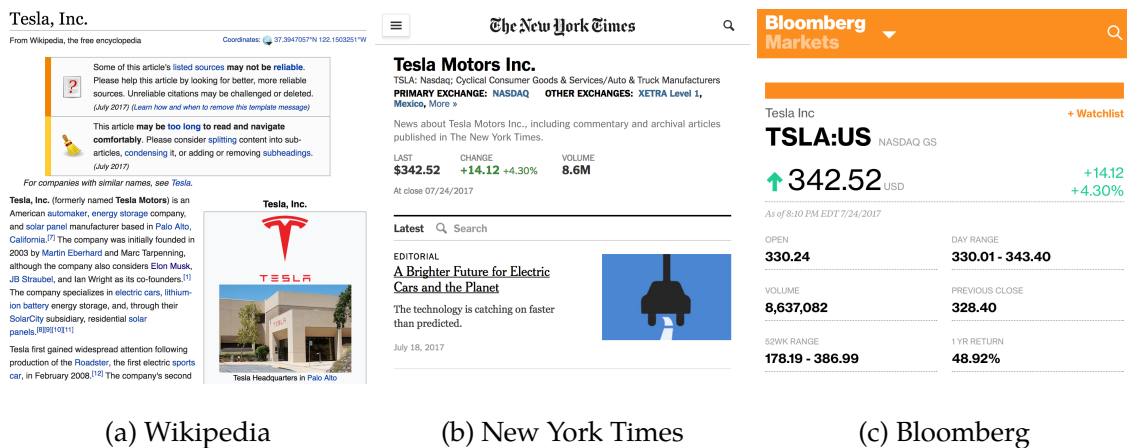


Figure 5.1: Three web pages representing the same Tesla Motors entity

As is the case for entity mentions in text, entity endpoints on the web are not particularly useful unless grounded by some common point of reference. Here again we face a problem of ambiguity resolution. While it may be possible to address endpoint ambiguity in a manner analogous to entity linking — by resolving references to a specific KB, we instead address a more general version of this problem by grouping coreferent endpoints into coherent clusters. In doing this, we address the problem of grounding URLs to an arbitrary KB by proxy — if a cluster contains links to a KB like Wikipedia, other endpoints in the cluster reference that entity by definition.

Identifying both positive and negative cases of coreference is critical to extending the respective depth and breadth of aggregated entity knowledge. In the example above, if we are able to correctly identify that these pages are coreferent, we can collectively exploit both the content of these web pages and the combined set of inlinks into them. On the other hand, true negative cases reveal a distinction between entities. If a candidate is non-coreferent with existing entity clusters, we may infer the existence of a new, previously unseen entity.

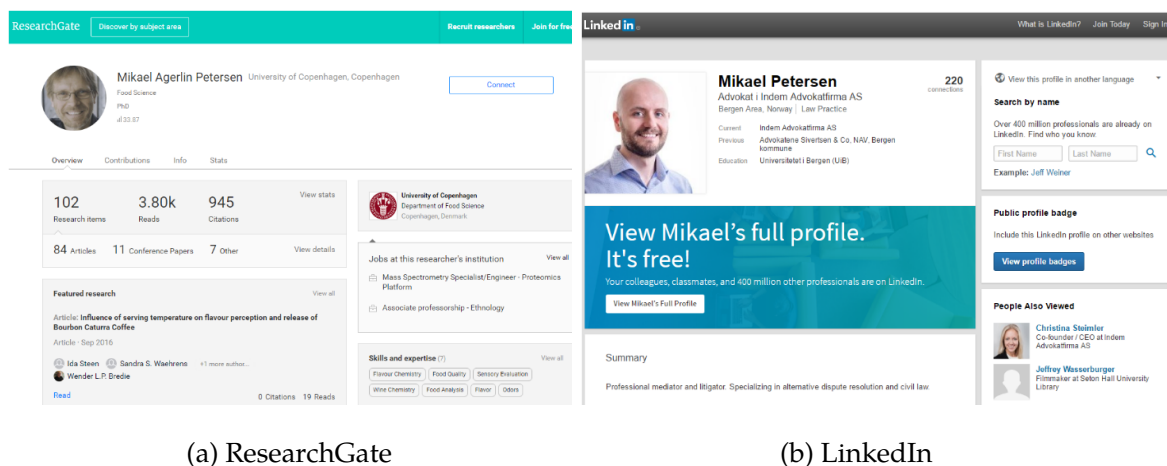


Figure 5.2: Two web pages representing distinct Mikael Petersen entities.

This chapter explores the task of clustering coreferent entity endpoints across discrete web KBs. Building on our entity disambiguation framework from Chapter 3, we once again model entities in terms of inbound links targeting each entity URL. We develop a set of features modelling pairwise coreference between mention clusters and train a supervised model over unlabeled documents leveraging corpus heuristics which generate instances of positive and negative coreference. We then apply this model to iteratively aggregate pairwise decisions into clusters of coreferent entity endpoints.

Experiments in this chapter address collections with orders of magnitude more documents and links than previously considered. Here we demonstrate that link-driven KBD and KB-Coref scale up to corpora which approximate the web as a whole by building on datasets from the CommonCrawl collection including over 1.5B documents

and 4B links. We evaluate end-to-end KBD and KB-Coref on these new corpora by annotating a sample of ambiguous entity pages and analyzing both the types of endpoints recovered through KBD and the results of end-to-end clustering baselines on this data.

5.2 Related work

Entity ambiguity is central to many natural language understanding tasks. This chapter addresses ambiguity amongst web entity endpoints at the KB level using inlinks as a source of entity knowledge. While this particular task configuration appears unique, many other ambiguity resolution tasks are closely related.

Tasks such as record linkage (Fellegi and Sunter, 1969; Xu et al., 2013) and entity alignment (Hao Zhu, 2017) which match instances across distinct KB schema are similar. As are linked open data (Bizer et al., 2008) initiatives¹ which focus on curation of cross-KB links and ontology matching (Euzenat and Shvaiko, 2007) systems which map instances across distinct semantic web ontologies. Our task formulation differs from these approaches along two key dimensions. First, while most alignment problems address pairwise coreference, we specifically target the *clustering* problem for a potentially large number of web KBs. More importantly, we address entirely unstructured entity representations (i.e. collections of natural language mentions) in place of structured database records or nodes within a knowledge graph. While some endpoint targets retain structured or semi-structured knowledge resources for an entity (e.g. those linking to traditional KBs), many still do not (e.g. pages aggregating news articles about an entity).

Entity linking systems (Bunescu and Paşca, 2006) which resolve ambiguous mentions in text to their corresponding node in a knowledge base are also related. Here ambiguous instances are resolved by modelling the similarity between a query instance and the structured representation of an entity stored in the KB. In Chapter 3 we develop

¹<http://linkeddata.org/>

NED models from unstructured web inlinks to the KB, but still rely on a single KB as a target for entity resolution. Hachenberg and Gottron (2012) consider an interesting variation on this task — searching the web to collect links which represent a KB entity, but do not consider the generalized task of clustering endpoints for any linked entity on the web. Web Person Search (WePS) (Artiles et al., 2005, 2010) presents a query oriented version of this task for person entities on the web. WePS takes the output of a web search for an entity name and attempts to cluster results that refer to the same underlying entity. While this formulation of the web endpoint coreference task is tailored to enriching web search results, our experiments in this chapter focus on a corpus level coreference. In addition, we utilize KBD to recover likely endpoint URLs directly from the target corpus, negating the need for proprietary web search in link discovery.

We developed a similar benchmark to the WePS task targeting entities at the long-tail of the notability distribution in Chisholm et al. (2016a). Building on the KBD system described in Chapter 4, we sampled rare entity names from entity endpoint links in the HG-NEWS corpus and search the web to generate a balanced dataset of ambiguous web page pairs. We ran this task at the 7th Australian Language Technology Association (ALTA) workshop in 2016. Of the 6 participating systems, the winning team EOD (Khirbat et al., 2016) achieved an F-score of 0.86 classifying pairwise coreference over a held out test set of 100 URL pairs. As is the case for WePS, we observe that systems primarily utilize the content of a target web page when resolving pairwise ambiguity. Subsequent work building on this task and dataset explores the use of distant supervision to recover additional entity endpoint links (Shivashankar et al., 2017). While these systems predominately leverage the content of an entity endpoint page in disambiguation, our work in this chapter models endpoints exclusively in terms of inbound links.

Cross-document coreference resolution (CDCR) (Bagga and Baldwin, 1998; Gooi and Allan, 2004) presents the closest task configuration to experiments described in

this chapter. CDCR systems resolve ambiguous entity mentions in text by clustering together coreferent mentions across documents in a corpus. While these systems generally operate over unstructured mentions and are not bound by the coverage of a fixed KB, global context from KB links has often been used as a source of knowledge for within-document coreference resolution (Ponzetto and Strube, 2006; Ratnov and Roth, 2012; Zheng et al., 2013). For large-scale cross-document coreference, Singh et al. (2011) present the closest work to our own. They construct a corpus of 1.5M entity mentions via inlinks to Wikipedia pages and develop a large-scale hierarchical clustering which groups coreferent entity mentions across pages in the corpus. While our experiments instead consider coreference across entity *endpoints*, we model each endpoints in terms of textual mentions from inbound links across the web. As such our task configuration closely resembles that of CDCR initialized by partially populated clusters. We additionally build a pair of web document datasets with KBD annotations representing a generalization of the Wikilinks corpus (Singh et al., 2012) to non-Wikipedia web KBs.

5.3 Methodology

In this section we provide a high-level overview of end-to-end experiments combining Web KBD and Cross-KB Coreference Resolution. Starting with a corpus of unlabeled documents from the web, we first train an improved KBD classifier. Here we augment both the features used and training methodology applied to improve endpoint recognition for problematic cases identified in the previous chapter. After running KBD, we aggregate inlinks for discovered endpoints into cluster of mentions for each discovered entity URL. Using these clusters, we are able to generate examples of positive and negative coreference across mention sets by sampling both within and across clusters respectively. We fit a supervised classifier on instances sampled from this data and use it to decide coreference between pairs of candidate URLs using textual context features

alone. Given this model for deciding pairwise coreference, we next turn to the task of building complete coreference clusters. To constrain our clustering problem, we only consider pairs of URLs which share an anchor text string in the corpus — i.e. entities with a common name. Under this constraint we enumerate all candidate URL pairs over which we must decide coreference to build coherent clusters. Whenever a pair of URLs are judged coreferent by the model, we merge their constituent inlinks together into a larger cluster of mentions. Our final system distributes and resolves independent pairwise coreference decisions in parallel, iteratively agglomerating URLs into clusters which each identify a distinct entity on the web.

5.4 Datasets

We conduct experiments using two datasets each built from web documents hosted by the CommonCrawl² project. CC-NEWS is derived from news articles crawled over a 6 month period from January to June 2017 inclusive. CC-WEB is derived from a snapshot of the entire web crawled in July 2017. Data from CommonCrawl is made available as WARC files which represent the full HTTP request and response of crawled web pages.

5.4.1 Preprocessing

For this data to be useful in our experiments, we must first perform significant preprocessing to recover plain-text documents and links. We first extract HTML document responses for successful web requests in the WARC corpus. We then perform language detection³ to filter non-English language documents. To mitigate the impact of outlier documents and parsing errors we also filter out documents beyond the 99.9th percentile in size (~250 KB). Next we extract plain text content from each HTML document using a machine learned content extraction library DragNet (Peters and Lecocq, 2013). This filters out text and links from non-content elements such as navigation menus and

²<https://commoncrawl.org>

³Chromium Compact Language Detector 2

advertisements on each page. We retain the set of links that appear inside textual content blocks and record both their URL target and in-document offset alongside the plain-text content of processed documents.

```

1  WARC/1.0
2  WARC-Type: request
3
4  ...
5
6  GET /2012/11/06/nicki-minaj-promises-man-bits-on-her-upcoming-tour/ HTTP/1.0
7  Host: 1019ampradio.cbslocal.com
8  Accept-Encoding: x-gzip, gzip, deflate
9  User-Agent: CCBot/2.0 (http://commoncrawl.org/faq/)
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
11
12 WARC/1.0
13 WARC-Type: response
14 WARC-Date: 2017-07-20T12:34:34Z
15 WARC-Record-ID: <urn:uuid:55843eb1-08c3-4060-867b-4933d0393447>
16 Content-Length: 114501
17 Content-Type: application/http; msgtype=response
18 WARC-Warcinfo-ID: <urn:uuid:2e3b6c25-24c0-4c52-a814-6fb1510e2786>
19 WARC-Concurrent-To: <urn:uuid:562e7ef4-4ca2-4f98-aaf9-115ced1db0bd>
20 WARC-IP-Address: 192.0.79.33
21 WARC-Target-URI: http://1019ampradio.cbslocal.com/2012/11/06/nicki-minaj-promises
    -man-bits-on-her-upcoming-tour/
22 WARC-Payload-Digest: sha1:H4TAABSMY7AQZ5SN2ZFWZGOKGY5SOA2B
23 WARC-Block-Digest: sha1:MKJZWBEPJ5IBBJSEUWBBVX07PMKWOY2B
24 WARC-Truncated: length
25 WARC-Identified-Payload-Type: text/html
26
27 HTTP/1.1 200 OK
28 Server: nginx
29 Connection: close
30 Date: Thu, 20 Jul 2017 12:34:34 GMT
31 X-Pingback: http://1019ampradio.cbslocal.com/xmlrpc.php
32 Vary: Cookie
33 X-hacker: If you're reading this, you should visit automattic.com/jobs and apply
    to join the fun, mention this header.
34 Link: <http://wp.me/p2qyBV-pHK>; rel=shortlink
35 Content-Type: text/html; charset=UTF-8
36 X-ac: 4.dca _dca
37
38 <!DOCTYPE html>
39 <html lang="en">
40 <head>
41     <meta charset="UTF-8" />
42     <title>Nicki Minaj Promises Man Bits On Her Upcoming Tour &laquo; 101.9
        AMP Radio</title>
43     <meta name="description" content="&quot;Maybe we&#039;ll have some flying
        penises. Imagine we just have little penises flying through the air,&
        quot; she said." />
44     <link rel="pingback" href="http://1019ampradio.cbslocal.com/xmlrpc.php" />
45
46     <meta name="keywords" content="vibNews" />
47 ... <!-- truncated:114KB total -->

```

Listing 5.1: Sample WARC encoded input from the CommonCrawl corpus before preprocessing. Records include request metadata and HTML encoded page content.

```

1 {
2   "_id": "http://1019ampradio.cbslocal.com/2012/11/06/nicki-minaj-promises-man-
      bits-on-her-upcoming-tour/",
3   "text": "Nicki Minaj has had quite the year. Currently in the U.K. on her
      Reloaded Tour she sat down with London DJ Tim Westwood and her U.K. Barbz
      for a Q & A session. While Nicki took questions from both Westwood and her
      fans one answer in particular caused the room to pay attention...",
4   "links": [{
5     "start": 0,
6     "endpoint": 0.6358972797,
7     "stop": 11,
8     "target": "http://1019ampradio.cbslocal.com/tag/nicki-minaj"
9   }, {
10    "start": 145,
11    "endpoint": 0.2769776554,
12    "stop": 160,
13    "target": "http://www.youtube.com/watch?feature=v=vnyuhDBcQo0"
14  }],
15  "mentions": [{
16    "start": 0,
17    "stop": 11,
18    "label": "PERSON"
19  }, {
20    "start": 53,
21    "stop": 57,
22    "label": "GPE"
23  },
24  // truncated
25 }

```

Listing 5.2: Sample JSON encoded document from the processed CC-WEB corpus. Output includes the URL of the crawled page, extracted text and recorded offsets of outlinks and named entity mentions. endpoint attributes on each link represent the probability assigned by KBD that the link represents an entity endpoint URL.

Given the size of our target and parallelizable nature of the task, we utilize a distributed extraction pipeline built on Apache Spark to preprocess the data. For the largest dataset, we utilize a cluster of 64 Amazon EC2 instances⁴ with 2048 cores which is able to process the 63 TB corpus in ~20 hours of wall-clock compute time.

⁴Compute Optimized — c3.8xlarge

Data set	Size (raw)	Size	Documents	Links	Words
HG-NEWS	n/a	4 G	3 M	14 M	1690 M
CC-NEWS	873 G	47 G	14 M	28 M	2209 M
CC-WEB	63,354 G	997 G	1,565 M	4,194 M	385 B

Table 5.1: Summary statistics for each dataset.

5.4.2 Statistics

Table 5.1 lists summary statistics for each corpus. We include statistics for the HG-NEWS corpus described in Chapter 4 for comparison, though we do not utilize it in experiments for this Chapter. Size (raw) refers to the size of the input corpus, while Size denotes the size of the corpus after preprocessing. In each case we list gzip-compressed corpus size.

CC-WEB represents a huge corpus by the standards of contemporary NLP. Even after filtering it contains more than 2 times as many documents as the ClueWeb12 collection. For tasks involving entities, the subset of each corpus with links targeting Wikipedia presents an interesting point of comparison. Table 5.2 compares inlinks to English Wikipedia from each web dataset with the Wikilinks corpus utilized in Chapter 3. Despite being older, Wikilinks includes approximately 39% *more* Wikipedia links than CC-WEB. Moreover, this comparison is likely conservative with respect to size of Wikilinks—Singh et al. (2012) additionally constrain their extraction to exclude pages duplicating Wikipedia content and links which do not match known aliases for a Wikipedia target. This difference suggests significantly lower overall coverage of pages linking to Wikipedia entities amongst the underlying CommonCrawl extraction. While the proprietary Google crawl index utilized by Wikilinks has significantly better coverage, we expect this gap to narrow over time. Subsequent CommonCrawl extractions have both significantly increased the crawl size (10-25%) and reduced the number of spam pages stored in the archive⁵. Additionally, our dependency on an open-source web crawl with a monthly

⁵ <http://commoncrawl.org/2017/11/november-2017-crawl-archive-now-available/>

	CC-NEWS	CC-WEB	Wikilinks
Links	37.9 K	29.0 M	40.3 M
Targets	69.1 K	2.7 M	2.9 M
Pages	35.3 K	9.8 M	10.9 M

Table 5.2: Comparison of links to English Wikipedia across web corpora.

release cycle simplifies the reproduction our experiments on updated snapshots of the web over time.

5.5 Identifying KBs

Before we can start clustering entity endpoints we first perform KBD to identify candidate entity URLs in each corpus. In addition to the basic KBD system described in 4, we augment our method in the following ways. First, we utilize an open source NER system⁶ to improve the reproducibility of our results.

We also attempt to address a number of problematic cases identified in the previous chapter by introducing entity identifier features and adjusting the way we sample training instances. Together these changes amount to a shift away from classifying aggregated endpoint patterns (e.g. `en.wikipedia.org/wiki/*`), to classification of individual endpoint URLs (i.e. `en.wikipedia.org/wiki/John_Smith`).

Under our original KBD model the feature representation for URLs which reduce to the same endpoint pattern is always the same. To differentiate between entities under the same endpoint, we add unigram features extracted from each URL entity identifier. Where the preceding URL path provides little evidence, these features are often informative. For example, the presence of an entity name in the target can indicate an entity reference (e.g. "John" in `twitter.com/john-smith`). These features also help distinguish between targets when the underlying endpoint indexes both entity and non-entity URLs (e.g. `wired.com/tag/tesla` and `wired.com/tag/electric-vehicles`). In cases

⁶ <https://spacy.io/> — version: 1.9; model: `en_core_web_sm`

Data set	P	R	F
CC-NEWS	0.50	0.51	0.50
CC-WEB	0.62	0.39	0.48

Table 5.3: Dev set performance for each KBD classifier on the mention prediction task.

where the identifier carries little semantic weight (e.g. numeric identifiers, product codes, GUIDs) they are however unlikely to be useful.

To account for the addition of these features, we make a corresponding adjustment to the way in which training instances are generated. Instead of aggregating links at the endpoint level, we aggregate inlinks by target URL before generating instances. For example, instead of generating a single instance for `en.wikipedia.org/wiki/<eid>`, we generate instances for `en.wikipedia.org/wiki/Tesla_Inc.` and `en.wikipedia.org/wiki/Self-governance` independently. As before, we generate a positive label when the majority of inlinks for an instance are mentions and negative otherwise. URLs for endpoints with fewer than 5 unique inbound URLs patterns are excluded.

5.5.1 Results

We train KBD models for both the CC-NEWS and CC-WEB datasets. Results for each model on the intrinsic mention prediction objective are described in Table 5.3. While mention prediction F -scores are lower than those described in the preliminary experiments of Chapter 4, here we aggregate model performance at the level of individual URLs, providing a better account of mention-prediction performance. A post-hoc evaluation of endpoint prediction in the context of the end-to-end KB-Coref task is described in Section 5.8.

After training, we select a confidence threshold and extract endpoint links from each dataset. In subsequent coreference experiments, we prefer high precision over endpoint recall. This both significantly reduces the number of coreference decisions required and improves the quality of the resulting endpoint clusters. We select a

Data set	Mentions	Endpoints	Patterns	Domains
CC-NEWS	6.37 M	1.27 M	0.17 M	0.62 M
CC-WEB	147.82 M	10.97 M	2.54 M	29.48 M

Table 5.4: Statistics of high-confidence entity links extracted from each dataset.

Endpoint	Entities	Description
leica-users.com/vNN	698,220	Mailing list archive
mesonet.agron.iastate.edu/gis	441,701	Geographic information system
filetransit.com/download	356,702	Index of software downloads
oemcats.com/oem-parts	297,907	Car components catalogue
nudipixel.net/photo/NNNNNNNN	250,680	Sea slug taxonomy
patentsencyclopedia.com/inventor	229,688	Record of patent holders
comicbookdb.com/issue	206,768	Catalog of comic books issues
thebaseballcube.com/players/profile	196,118	Directory of baseball players
artslant.com/global/artists/show	188,017	Artist biographies
reservations.airportguide.com/hotel	152,538	Hotel listings

Table 5.5: Sample of top-10 endpoint patterns by unique inlink count.

threshold $P(m|u) \geq 0.95$ and filter out links from each dataset which fall below this threshold. Table 5.4 details the resulting endpoint URL statistics extracted from each dataset after filtering. In counting domain names we consider all sub-domains excluding `www` as distinct and do not account for active redirection (e.g. URL shortening services such as `goo.gl` or `bit.ly`).

5.5.2 Analysis

Despite setting a high threshold in endpoint probability, we still recover a large collection of candidate endpoint patterns from each web dataset. Table 5.5 lists the top-10 endpoint patterns by unique inlink count in the CC-WEB dataset.

Encouragingly, we observe a great variety of high-quality knowledge base structure in classified links. Endpoints covering sporting teams and athletes for specific verticals

are common, e.g. `thebaseballcube.com/players` and `basketball-reference.com/teams`. We also find a wide variety of other interesting domains, e.g. from car part reference lists `oemcats.com/oem-parts` to sea-slug taxonomies `nudipixel.net/photo`.

Interestingly, the largest endpoint pattern by unique inlink count is a mailing-list archive `leica-users.com/vNN` for users of Leica cameras. Most URLs under this endpoint represents conversations around an email subject which may or may not represent an entity. For example, discussion of camera models (e.g. The Leica R8: `v00/msg03451.html`), famous photographers (e.g. Margaret Bourke-White: `v00/msg03122.html`) and photos taken of a location (e.g. Barcelona Cafe de l’Opera: `v58/msg15965.html`) uniquely identify specific real-world entities. However, in cases where these URLs do not represent distinct entities, the URL itself is uninformative. These instances suggest that content-based features may be critical to further improving KBD.

5.6 Resolving link coreference

After identifying entity URLs in our corpus, we now turn to the problem of resolving pairwise KB-Coref. When deciding if a pair of candidate URLs (a, b) are coreferent, we first need some model of the entities represented by a and b . In the web setting, systems may draw upon either the content of an entity’s web page, or the context of pages linking to that site across the web.

Endpoint pages often denote the name of the entity and may additionally index important disambiguating information such as dates of birth, occupation, interests or other descriptive text. Even where details of the entity itself are not described directly other contextual information may be present on the page, as is the case for news tag pages which reproduce article text referencing an entity. Content driven entity modelling is especially important for long-tail entities which otherwise accrue few mentions via inbound web links. For example, supervised learning over content-driven

similarity features was the predominant approach in the ALTA 2016 long-tail pairwise web coreference shared task (Chisholm et al., 2016a).

Where entity mentions are present in the form of inbound link for an entity URL, they present an alternate source of textual knowledge. While this kind of knowledge is less readily available for long-tail entities, experiments in Chapter 3 demonstrate it is a valuable source of disambiguating information. For systems targeting a diverse set of web KBs, consistent extraction of useful entity attributes from page content is challenging given variation in page structure across endpoints. By contrast, textual context from inlinks is an essentially homogeneous store of unstructured knowledge independent of the target KB schema. Moreover, in cases where the content of an endpoint page is temporarily unavailable, behind a pay-wall or otherwise access controlled⁷, we may still model a target entity through publicly accessible inbound web links.

While we expect both content and inlink driven entity modelling to be complementary in practice, we describe a simple and scalable representation for entities derived from aggregated inlinks to an entity URL. We extract mentions for each entity in our corpus by sampling a 3-sentence context window around anchors for endpoint links identified by KBD. We then develop a weakly-supervised classifier which predicts whether a given pair mention-sets reference the same underlying entity.

5.6.1 Entity representation

To represent an entity e in terms of linked mentions, we adopt a simple weighted bag-of-ngrams representation over link anchors and the surrounding textual context. From each sampled mention, we extract tokens and accumulate uni-gram and bi-gram term-frequencies for anchors and context separately.

This sparse representation presents two key advantages when scaling to large clusters of mentions per entity. First, by reducing mention and anchor context to term-frequency representations the size of each cluster grows in proportion to the number of

⁷This includes automated web-crawl restrictions imposed via robots.txt

unique terms rather than than sum of all tokens across aggregated mentions. This size may also be bounded via a simple feature hashing scheme. In addition, our eventual goal of agglomerative clustering entities requires that we merge representations as positive cases of coreference are identified across mention sets. In this case, we can simply perform term-wise addition of frequency counts to compute an equivalent merged feature representation for a pair of entity clusters. Without this property, we must rebuild the feature representation of a cluster from text as coreferent mentions are agglomerated — a comparatively expensive operation.

5.6.2 Features

For each instance of paired entity representations (a, b) , we aim to generate a small set of features that capture the similarity between these entities. We are motivated to select features which are both fast to compute and invariant to the ordering of entities a and b . We select simple similarity metrics analogous to those considered for ambiguity resolution in Chapter 3 and additionally compare the most-common anchor string for each mention set. The complete feature set includes:

- Cosine similarity over anchor token ngrams
- Cosine similarity over context token ngrams
- Cosine similarity over character ngrams for the most common anchors
- Binary feature for exact match between the most common anchors

The resulting representation encodes the similarity between a pair of sparse entity representations in just 4 dimensions. We illustrate this encoding through a constructed example in Figure 5.6.

URL	twitter.com/teslamotors	nytimes.com/topic/tesla-motors-inc
Mentions	.. announced by [Tesla] before bought a [tesla] Model X.. .. [Tesla Motors] makes over cars from [TSLA] are as [TSLA] trades lower before [Tesla Inc] released the [carmaker] announced while [Tesla Motors] cars ..
Counts	anchor:tesla=3 anchor:tesla_motors=1 text:model_x=1 text:cars=1 ...	anchor:tesla=2 anchor:tesla_motors=1 text:announced=1 text:cars=1 ...
Features	[0.6, 0.85, 1.0, 1.0]	

Table 5.6: Illustration of mentions and computed features for a candidate pair.

Mentions depicts the set of inlinks for each URL. **Counts** shows the derived sparse bag-of-ngrams representation of each mention set. **Features** represents the final vector of anchor similarity, context similarity, top-anchor match and top-anchor character similarity computed for the pair. Values in this table are constructed for illustration.

5.6.3 Instance sampling

In place of a hand-labeled set of training instances, we explore a simple sampling method for generating weakly-supervised training instances for KB-Coref classification. Using our corpus of documents with classified entity endpoint links, we first group together mentions targeting the same endpoint URL. We then leverage the implicit coreference amongst inlinks for each endpoint to sample instances of positive and negative coreference.

To generate positive instances, we randomly split groups with more than one mention into two discrete sets with 25-75% of the total mentions each. Given all mentions for the same target URL represent mentions of the same underlying entity, any two subsets sampled from instances in a group represent positive examples of mention set coherence. To generate negative instances, we randomly sample mention sets for two different target URLs in the corpus. In contrast to our strategy for sampling positive instances, we tolerate a small chance for erroneously sampling false-negatives when randomly sampling another URL from the corpus. To reduce this chance, we constrain our sample to exclude URLs with similar tokens in their path terminator. This filters potential pairs such as `en.wikipedia.org/wiki/Tesla,_Inc.` and `nytimes.com/topic/company/tesla-motors-inc` but cannot account for all variation in coreferent URL pairs, e.g. `bloomberg.com/quote/TSLA:US`.

5.6.4 Training the model

We train a random forest classifier by sampling instances of entity pairs and computing pairwise feature representations as described above. Before sampling instances we filter the training set to include only high-confidence entity endpoint URLs — i.e. those with a entity probability ≥ 0.99 as predicted by KBD. This constrains the size of the training set and reduces noise from non-entity endpoint link comparisons. For every URL in the input corpus, we re-sample up-to 4 subsets of instances to generate positive

Data set	Mentions	Entities	Positives	Negatives
CC-NEWS	3,966,013	953,251	854,172	1,453,638

Table 5.7: Training set statistics after filtering. Positives is the number of positive coreference pairs generated and Negatives is the number of negative pairs generated after 4-iterations of sub-sampling.

pairs and up-to 4 random contrastive URLs to generate negatives pairs. Table 5.8 shows statistics of the dataset after filtering.

We randomly sample 10% of instances by URL from the CC-NEWS dataset as a held-out development set to evaluate model performance. In contrast to the KBD task in which our model is conditioned on the links present in the corpus, our KB-Coref model is essentially independent. As such, we need not train a corresponding coreference classifier over the larger CC-WEB dataset. Coreference results described in the rest of this chapter utilize the coreference model trained on pairs sampled from the CC-NEWS dataset alone.

Our trained KB-Coref classifier achieves close to perfect results over development set instances. We observe an F score of 0.991 at a threshold of 0.5 and overall Area under PR curve of 0.994. These results suggest that distinguishing positive instances from randomly sampled negatives is an essentially trivial task for our model under this feature set. Improvements to the model or feature set are unlikely to yield significant performance gains in this setting. While a high level of performance on randomly sampled negatives is necessary for robust coreference resolution, it is by no mean sufficient. We address alternative instance sampling and supervision strategies as part our discussion of future work in Section 5.10.1. In the next section, we utilize our trained pairwise KB-Coref classifier for end-to-end clustering and evaluate performance over truly ambiguous coreference pairs.

5.7 Clustering

After training a model to decide coreference given a pair of mention sets, we can now begin aggregating together mentions sets across entity endpoints. As we have on the order of millions of entity URLs to cluster, considering all possible pairwise combinations is both inefficient and generally intractable.

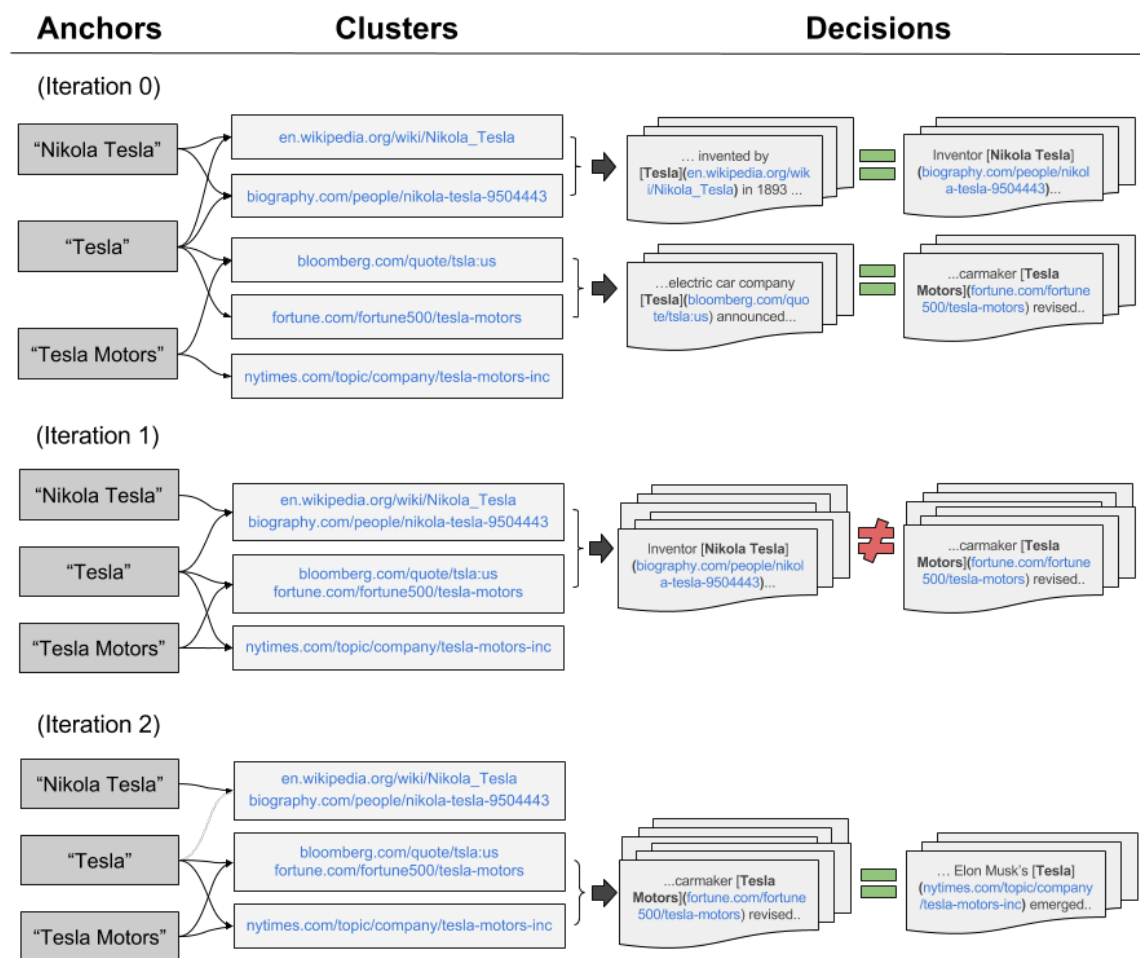


Figure 5.3: Iterative cluster aggregation of through pairwise mention set comparison.

5.7.1 Constraints

To reduce the size of our clustering problem, we impose the following constraints. First, we only consider a pair potentially coreferent if they share a common anchor

text string in the corpus. This does not rule out *eventually* clustering entity endpoints linked via alternative names. Consider three endpoints x , y and z with anchors $\{a\}$, $\{a, b\}$ and $\{b\}$ respectively. As long as we correctly identify coreference between the pairs (x, y) and (y, z) , we obtain (x, z) via transitive equality. We also constrain clusters to contain at most one URL instance for a given endpoint URL pattern. As KBs typically maintain a single canonical entry for an entity, we need not consider conference amongst candidate URL pairs under a common KB. For example, there should only be a single article representing Tesla Motors on Wikipedia. This constraint is analogous to the one word-sense per discourse heuristic described by Gale et al. (1992).

To reduce the impact of noisy anchors on candidate clusters, we also attempt to filter links to KB endpoints which do not represent mentions of the target entity, e.g. anchors such as "read more", "expand", "venue description" or "here". These anchors violate our assumption that anchors specify entity names and contribute noise to the clustering task. While many of these anchors are non-named entities and could be excluded via NER, many simply represent different named entities to the addresses target, i.e. those identifying the target site instead as in "facebook", "imdb" or "twitter". We adopt an ad-hoc data driven approach to filter the most prominent anchors of this type. We samples instances of anchor strings which reference 10 or more distinct entities across more than 15 distinct KBs. These bounds respectively set an upper limit on the level of ambiguity we expect to see for a given name per KB, and set a lower limit on the number of times we must observe a highly ambiguous anchor before filtering it. We manually explore alternative thresholds with a goal of minimizing false-positive anchors in the list. In total we identify 46 anchors under this criteria. All links with an anchor in this set are excluded from candidate clusters.

5.7.2 Iterative URL aggregation

After identifying possible pairs under our constraints, we classify them using our pairwise coreference model. Following a similar approach to Singh et al. (2011), we aim to speed up clustering by evaluating all independent coreference decisions in parallel. We detail this process in Figure 5.3. At each iteration we decide on at most one coreference decision per cluster. For each group of candidates to be clustered in an anchor set, we prioritize the URLs which have the highest inlink count first — i.e. the endpoints for which we have the best information. This approach is analogous to systems in entity disambiguation which aim to resolve the highest confidence decisions first and thereby improve relatedness measures for subsequent decisions (Milne and Witten, 2008). After each iteration, mention sets for pairs that are classified as positively coreferent are combined — decreasing the number of total clusters and increasing the average number of mentions per URL. This process is repeated until there are no more decisions to resolve under our constraints.

Algorithm 1 Iterative URL aggregation

Require: $n \geq 0 \vee x \neq 0$

Ensure: $y = x^n$

$C \Leftarrow$ endpoint clusters

$D \Leftarrow$ the set of decided endpoint pairs

repeat

$U \Leftarrow$ the set pairs to decide

until $|U| = 0$

Worst-case complexity for this approach is $O(n^2)$ iterations, where n is the size of the largest anchor set in the corpus and all candidate pairs are decided in parallel at each iteration. While this assumes each inlink in the set is non-coreferent, in general we expect most links in a candidate set to reference the same entity — following a Zipf-like distribution where inlink count is inversely proportional to rank. This is

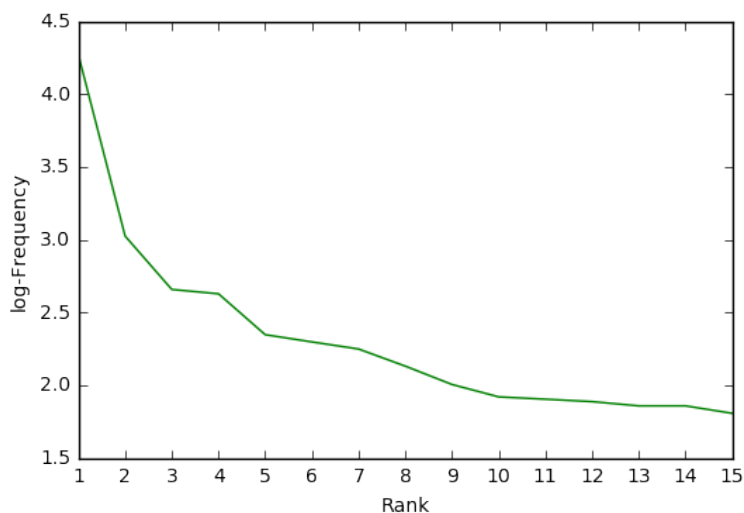


Figure 5.4: Frequency vs. Rank for link targets referencing “Obama” in their anchor text string across Wikipedia.

a consequence of the skewed entity mention frequency distribution — even when conditioned on an anchor span, the most notable entity for a name will account for almost all outbound URLs. For example, almost all links anchored by "Obama" will be mentions of the former U.S. president Barack Obama, so sampling from these links returns predominately positive instances of coreference. Figure 5.4 shows this relationship between inlink frequency and rank for links with Obama in the anchor across Wikipedia. The most common referent `en.wikipedia.org/wiki/Barack_Obama` aggregates 100x more inlinks than the entity at rank 10. This logarithmically reduces the number of expected comparisons per cluster, suggesting an average case complexity closer to $O(\log^2 n)$.

For the larger CC-WEB dataset, we must still make a number of algorithmic approximations to mitigate problems which arise from memory and processing constraints. At each iteration, we cap the number of coreference decisions by randomly sampling at most 1 M ambiguous pairs from candidate clusters to resolve. For some name clusters, we observe a huge number of targets (i.e. on the order of 100k) per cluster. To avoid deciding all possible combinations in a single iteration, we randomly shuffle targets

Data set	Anchors	URLs	URLs per Anch. (σ)	Iters	Decisions	Clusters
CC-NEWS	0.43 M	0.95 M	2.42 (18)	20	3.9 M	0.71 M
CC-WEB	4.51 M	10.97 M	2.78 (309)	50	364.5 M	6.97 M

Table 5.8: Number of input URLs and output clusters for each corpus.

and enumerate combinations in batches of at most 100 targets at a time. These approximations maintain a uniform likelihood for any given candidate pair to be decided at each iteration while mitigating combinatorial bottlenecks and bounding memory requirements.

5.8 Evaluation

In this section we discuss our evaluation results for KBD and KB-Coref on the CC-NEWS corpus. With no existing ground truth clustering of URLs available, we face a challenge in evaluating the clustering produced by our system. For standard cluster evaluation metrics, we must fully enumerate the set of instances which belong in a given gold standard cluster. In KB-Coref, this requires an exhaustive search the dataset for each sampled entity to identify every potential coreferent URL. This approach is laborious and disproportionately distributes annotation effort towards larger clusters of notable entities.

We instead opt to measure performance through a sampled evaluation across candidate coreferent pairs. We randomly sample first a cluster of endpoint URLs which share a common anchor text string then sample and annotate a pair of links within the cluster. Here we assess both the type of page referenced by each URL (i.e. to evaluate KBD classifications) and whether each page references the same underlying entity (i.e. to assess KB-Coref system clustering). We repeat this sampling strategy until we obtain 500 annotated pairs where both URLs represent valid entity endpoints.

	KB Entry	Entity Tag	Non-entity	Invalid
Count	820	411	175	114
Percentage	53.9%	27.1%	11.5%	7.5%

Table 5.9: Distribution of entity-link types extracted by KBD over annotated samples.

5.8.1 Endpoint results

For each URL we consider during annotation, we categorize the targeted page as either a KB entry, entity tag page, non-entity reference or invalid link. Table 5.9 details the distribution of link categories observed during annotation. We describe each category in detail as follows:

KB entries Pages which aggregate original content describing an entity. These pages often have an entity description, photograph or other statistics providing a rich source of entity knowledge. Examples which fit into this category include high quality endpoints such as Wikipedia and biography.com/people. We consider these links true positives under our KBD evaluation.

Entity tags Lower quality pages which do not contain original content, but still uniquely identify an entity. News sites often contain endpoints of this type, where each page represents a tag aggregating references to an entity across articles on the website. As above, we consider these links true positives when evaluating KBD.

Non-entity pages Includes targets which do not reference a specific named entity (e.g. "electric vehicles" or "terrorism") or do not represent entity endpoints (e.g. a news article referencing an event involving the entity). These links represent false positives under our KBD evaluation.

Invalid links Includes pages which cannot be accessed or are otherwise invalid. We observe some links which require user authentication, exist behind a pay-wall or

Clustering	P	R	F
Anchor match	78.2	100.0	87.8
Classifier	82.4	90.0	86.0

Table 5.10: Clustering metrics for each baseline over sampled gold-standard clusters.

have simply gone stale due to changes in site structure without redirection. These are ignored for the purpose of our KBD evaluation.

We manually annotate a total of 1,520 individual URLs before reaching our goal of 500 pairs where both pages represent valid entity endpoints. Over the subset of valid links, 1,231 links in total are true positive entity endpoints, representing a precision of 87.5% over valid URLs for our KBD classifier.

5.8.2 Clustering results

After annotating a sample of valid endpoint pairs, we now evaluate whether coreferent URLs appear together in the final clustering of alternative systems. Table 5.10 details precision, recall and F-score over the 500 annotated URL pairs for each configuration. Under our evaluation scheme, precision is an indicator of cluster homogeneity and recall is a measure of cluster completeness. We denote significant results in bold where the score lies outside the 95% confidence interval of compared metrics for each system. Confidence intervals are calculated by bootstrapped re-sampling over 10,000 iterations.

Anchor match represents a system where all URLs which share a common anchor are clustered together. **Classifier** represents the results of agglomerative clustering driven by our weakly-supervised classifier. As previously observed, over a natural distribution of entities most mentions for a given name will be references to the same entity. Here we observe the same relationship holds for URLs which reference an entity on the web. This characteristic yields a high-bar for precision in the name-match system despite a 100% recall. Notably, our evaluation does not however account for coreference across endpoints which never share an anchor span in common. By

comparison our model driven KB-Coref clustering system achieves a significantly lower recall. This clustering is however able to improve upon the precision of a naive solution. In the following section, we conduct a qualitative analysis of clusters produced by our KB-Coref model and characterize the main types of errors observed.

5.9 Analysis

Table 5.11 lists a sample of URLs from clusters which aggregate entity pages for the Tesla Motors and Nikola Tesla entities. In this instance, the model is able to clearly distinguish URLs referencing each entity despite a common name reference.

In many cases however, we note precision errors where URLs for related entities have been incorrectly grouped into a single cluster. For example, the Telsa Motors cluster referenced in Table 5.11 also contains some references to URLs which represent cars produced by the company `leftlanenews.com/new-car-buying/tesla/model-x` and non-entity URLs which are topically similar `mirror.co.uk/all-about/electric-cars`. In addition to precision errors, we also find instances of other smaller clusters referencing the same URL which have not been aggregated together. For example, the Wikipedia endpoint for Tesla Motors appears in a smaller secondary cluster alongside `business.financialpost.com/tag/tesla-inc` and `economictimes.indiatimes.com/topic/tesla-inc`. In this case, encyclopedic descriptions of the entity may differ enough from popular news coverage to prevent aggregation by the model.

Tesla Motors	Nikola Tesla
<code>androidcommunity.com/tag/tesla</code>	<code>dailycollegian.com/tag/nikola-tesla</code>
<code>bloomberg.com/quote/tsla:us</code>	<code>biography.com/people/nikola-tesla-9504443</code>
<code>nytimes.com/topic/company/tesla-motors-inc</code>	<code>en.wikipedia.org/wiki/nikola_tesla</code>
<code>fortune.com/fortune500/tesla-motors</code>	<code>mysteriousuniverse.org/tag/nikola-tesla</code>

Table 5.11: Clusters for the Tesla Motors and NikolaTesla entities in the CC-NEWS dataset. URLs from the Tesla Motors cluster have been truncated.

Small clusters also appear to be more homogeneous. In larger clusters, we observe a tendency towards incorrectly accumulating multiple references for one or more less notable entities. For example, in a cluster with 65 URLs referencing U.S. president Donald Trump we observe a small sub-cluster of 8 URLs referencing the fashion designer Tommy Hilfiger. This may be representative of a kind of semantic drift (Curran et al., 2007) where once a single entity URL is incorrectly assigned to a cluster, subsequent additions for the entity are far more likely to be incorrectly aggregated together.

5.10 Discussion

In this chapter we investigate the end-to-end web entity discovery and coreference resolution task. Starting with raw WARC encoded request data from the CommonCrawl corpus, we extract plain-text page content and outbound web links. We then NER tag the text and train an improved KBD system to infer the presence of URLs which represent entities. Finally, we develop a coreference classifier over inbound links and use it to iteratively aggregate entity links into coreference clusters which each represent a distinct entity on the web. Despite observing overall lower performance for our weak-supervised clustering baseline, our qualitative evaluation of produced clusters is encouraging. Our KB-Coref system is able to aggregate coreferent links with reasonable accuracy, often from unexpected sources (e.g. `comicbookdb.com`) and often over entities which do not otherwise appear in a structured KB like Wikipedia (e.g. `/issue.php?ID=418932`). Still, the scale and variety of content on the web present numerous challenges. To retain quality output and computationally tractability we significantly constrain the output of KBD and make simplifying assumptions in KB-Coref which likely reduce coreference recall.

In addition to the developed baselines and clustering evaluation, we expect the distribution of the CC-WEB corpus with over 1.5B plain-text documents including stand-off annotation for NER, web links and KBD probabilities has great potential for

use across a variety of NLP and IE tasks. In contrast to other large NLP datasets, the ability to update the corpus on a monthly basis via compatibility with open-access CommonCrawl data enables ongoing entity information extraction for applications where currency is critical.

5.10.1 Future work

Experiments in this chapter suggest multiple clear directions for improving end-to-end entity endpoint discovery and Cross-KB coreference resolution. For KBD, we observe many instances where information present in the URL alone is clearly insufficient to infer the presence or absence of an entity reference. In these cases, utilizing information from the content of the page itself is likely critical to improving KBD performance. However, parsing and classifying the content of a web-page will significantly increase the complexity and run-time of models which already incur a large cost in targeting web-scale corpora.

For KB-Coref, weak supervision via randomly sampled negatives is likely insufficient to fit a robust coreference model over truly ambiguous pairs. For example, randomly sampled negatives will rarely share a common name, while most cases of true negative coreference at test time do. While we are able to improve upon the precision of a weak clustering baseline, training a model over non-trivial cases of coreference may improve end-to-end clustering performance. Specifically, we suggest taking advantage of a one entity identifier per KB heuristic. If a pair of links share an common anchor but reference *distinct* targets under the same endpoint, they present a non-trivial case of negative coreference. For example, consider links to both `en.wikipedia.org/w/John_Smith_(painter)` and `en.wikipedia.org/w/John_Smith_(policitian)` that share the anchor text "John Smith". To the extent that web KBs follow the pattern of unique targets for each covered entity, sampling negatives in this manner may provide a more representative corpus for weakly supervised coreference resolution. In addition

the integration of content-based features over web page targets and features of the candidate URLs themselves are likely to further improve clustering performance.

5.11 Summary

KBD and KB-Coref together present a mechanism for discovering and aggregating entity endpoints on the web. Web KBs offer a broader range of entity coverage in comparison to standalone knowledge stores, though clear challenges remain in accurately clustering coreferent endpoints across discrete KBs. Our core contribution is the construction of two web document collections and a preliminary evaluation of coreference clustering on these corpora — laying the groundwork for applications of large-scale knowledge extraction from up-to-date web resources.

In the following chapters, we build on this work by investigating models which translate information between the unstructured natural language forms prevalent on the web and more structured entity knowledge representations. In so doing, we aim to bridge the gap between the resources provided by web KBs and those inherent to a traditional structured knowledge store.

6 Biography generation

The art of writing is the art of
discovering what you believe.

Gustave Flaubert

We have so far considered methods by which entity knowledge may be aggregated from link structures prevalent on the web. In the ideal case, these methods deliver a large corpus of text unambiguously linked to a diverse set of entities. However, as a store of entity information, linked text alone is often insufficient in downstream tasks which apply entity knowledge.

In the final chapters of this thesis we attempt to bridge the functional gap between knowledge aggregated through web KBs and that available in a traditional structured knowledge store. We consider two typical components of a curated KB. First, natural language descriptions which summarize available entity information for human consumers. And later, structured facts which more often find application in automated systems for search, categorization and question answering.

In this chapter we take as given a factual representation of an entity and attempt to generate a concise textual description. We address this task as one of **knowledge translation** — leveraging the insight that facts and text describing an entity are distinct but often equivalent representations of entity knowledge. Our experiments suggest generated descriptions are comparable to a human written reference in terms of readability, though we observe a tendency for the model to infer and express facts which may not be explicitly present in the input.

Our contributions include: (1) a sequence-to-sequence translation model for fact driven entity description generation; (2) detailed development experiments and analysis of evaluation measures for fact driven text generation; Experiments detailed in this chapter were first described in Chisholm et al. (2017). Code and data for translation experiments are available at: github.com/andychisholm/eacl17gen.

6.1 Introduction

KBs like Wikipedia maintain a canonical natural language description for each entity. These descriptions aim to concisely convey the most salient facts about an entity in a format which is easily consumed by human readers. While descriptions have great utility, they require dedicated human effort to maintain. As facts about an entity change over time, the encoding of this knowledge in both the textual summary and structured store may become decoupled from each other and the underlying ground truth, imposing an ongoing burden in knowledge curation. Despite these costs, textual summaries are clearly central to the value of a KB for human consumers.

We explore the task of generating entity descriptions from factual knowledge. We focus on generating one-sentence biographies for human entities in the English Wikipedia using facts from Wikidata. Figure 6.1 shows a Wikidata entry for an example squash player Mathias Tuomi, with fact keys and values flattened into a sequence alongside the first sentence from his Wikipedia article. Some values are in the text, others are missing (e.g. male) or expressed differently (e.g. dates).

We treat this *knowledge-to-text* task like translation, using a recurrent neural network (RNN) sequence-to-sequence model (Sutskever et al., 2014) that learns to select and realise the most salient facts as text. This includes an attention mechanism to focus generation on specific facts, a shared vocabulary over input and output, and a multi-task autoencoding objective for the complementary extraction task. We create a reference dataset comprising more than 400,000 knowledge-text pairs for person enti-

```
TITLE mathias tuomi SEX_OR_GENDER  
male DATE_OF_BIRTH 1985-09-03  
OCCUPATION squash player  
CITIZENSHIP finland
```

Figure 6.1: Example Wikidata facts encoded as a flat input string. The first sentence of the Wikipedia article reads: Mathias Tuomi, (born September 30, 1985 in Espoo) is a professional squash player who represents Finland.

ties, handling the 15 most frequent slots. We also describe a simple template baseline for comparison on BLEU and crowd-sourced human preference judgements over a heldout TEST set.

Our model obtains a BLEU score of 41.0, compared to 33.1 without the autoencoder and 21.1 for the template baseline. In a crowdsourced preference evaluation, the model outperforms the baseline and is preferred 40% of the time to the Wikipedia reference. Manual analysis of content selection suggests that the model can infer knowledge but also makes mistakes, and that the autoencoding objective encourages the model to select more facts without increasing sentence length. The task formulation and models are a foundation for text completion and consistency in KBs.

6.2 Related work

RNN sequence-to-sequence models (Sutskever et al., 2014) have driven various recent advances in natural language understanding. While initial work focused on problems that were sequences of the same units, such as translating a sequence of words from one language to another, other work been able to use these models by *coercing* different structures into sequences, e.g., flattening trees for parsing (Vinyals et al., 2015b), predicting span types and lengths over byte input (Gillick et al., 2016) or flattening logical forms for semantic parsing (Xiao et al., 2016).

RNNs have also been used successfully in *knowledge-to-text* tasks for human-facing systems, e.g., generating conversational responses (Vinyals and Le, 2015), abstractive summarisation (Rush et al., 2015). Recurrent LSTM models have been used with some success to generate text that completely expresses a set of facts: restaurant recommendation text from dialogue acts (Wen et al., 2015), weather reports from sensor data and sports commentary from on-field events (Mei et al., 2015). Similarly, we learn an end-to-end model trained over key-value facts by flattening them into a sequence.

Choosing the salient and consistent set of facts to include in generated output is also difficult. Recent work explores unsupervised autoencoding objectives in sequence-to-sequence models, improving both text classification as a pretraining step (Dai and Le, 2015) and translation as a multi-task objective (Luong et al., 2016). Our work explores an autoencoding objective which selects content as it generates by constraining the text output sequence to be predictive of the input.

Biographic summarisation has been extensively researched and is often approached as a sequence of subtasks (Schiffman et al., 2001). A version of the task was featured in the Document Understanding Conference in 2004 (Blair-Goldensohn et al., 2004) and other work learns policies for content selection without generating text (Duboue and McKeown, 2003; Zhang et al., 2012; Cheng et al., 2015). While pipeline components can be individually useful, integrating selection and generation allows the model to exploit the interaction between them.

KBs have been used to investigate the interaction between structured facts and unstructured text. Generating textual templates that are filled by structured data is a common approach and has been used for conversational text (Han et al., 2015) and biographical text generation (Duma and Klein, 2013). Wikipedia has also been a popular resource for studying biography, including sentence harvesting and ordering (Biadys et al., 2008), unsupervised discovery of distinct sequences of life events (Bamman and Smith, 2014) and fact extraction from text (Garera and Yarowsky, 2009). There has also been substantial work in generating from other structured KBs using tem-

plate induction (Kondadadi et al., 2013), semantic web techniques (Power and Third, 2010), tree adjoining grammars (Gyawali and Gardent, 2014), probabilistic context free grammars (Konstas and Lapata, 2012) and probabilistic models that jointly select and realise content (Angeli et al., 2010). As an alternative to sequence-to-sequence models, recent work also explores the application of Variational AutoEncoders (VAEs) (Kingma and Welling, 2013) to the text domain (Bowman et al., 2016). Recently Novikova et al. (2017) publish an evaluation dataset for end-to-end generation models, targeting mappings between meaning representations (i.e. fact-value pairs) and restaurant descriptions. Subsequent systems demonstrate the surprising effectiveness of character level sequence-to-sequence generation (Agarwal and Dymetman, 2017) on this task.

Lebret et al. (2016) present the closest work to ours with a similar task using Wikipedia infoboxes in place of Wikidata. They condition an attentional neural language model (NLM) on local and global properties of infobox tables, including *copy actions* that allow wholesale insertion of values into generated text. They use 723k sentences from Wikipedia articles with 403k lower-cased words mapping to 1,740 distinct facts. They compare to a 5-gram language-model with copy actions, and find that the NLM has higher BLEU and lower perplexity than their baseline. In contrast, we utilise a deep recurrent model for input encoding, minimal slot value templating and greedy output decoding.

Vougiouklis et al. (2017) also consider the task of Wikipedia biography generation. Their model embeds multiple relational triples under a single fixed-length vector representation as input for a RNN decoder network similar to our own. In addition to Wikidata, they consider facts derived from DBpedia and extend their evaluation to the first two sentences of each Wikipedia article. As they do not isolate first sentence generation performance and sample a different subset of entities their results are not directly comparable to our own.

Evaluating generated text is challenging and no one metric seems appropriate to measure overall performance. Lebret et al. (2016) report BLEU scores (Papineni et al.,

2002) which calculate the n-gram overlap between text produced by the system with respect to a human-written reference. Summarisation evaluations have concentrated on the content that is included in the summary, with semantic content typically extracted manually for comparison (Lin and Hovy, 2003; Nenkova and Passonneau, 2004). We draw from summarisation and generation to formulate a comprehensive evaluation based on automated metrics and human validation. Our final system comparison follows (Kondadadi et al., 2013) in running a crowd task to collect pairwise preferences for evaluating and comparing both systems and references. Notable subsequent work includes the WebNLG task (Gardent et al., 2017) in which systems map a set of RDF triples to a textual description for an entity from any of 9 diverse and distinct DBpedia categories (e.g. sports teams, universities, buildings, food and others).

6.3 Task and data

We formulate the one-sentence biography generation task as shown in Figure 6.1. Input is a flat string representation of the structured data from the KB, comprising slot-value pairs (the subject being the topic of the KB record, e.g., Mathias Tuomi), ordered by slot frequency from most to least common. Output is a biography string describing the salient information in one sentence.

We validate the task and evaluation using a closely-aligned set of resources: Wikipedia and Wikidata. In addition to the KB maintenance issues discussed in the introduction, Wikipedia first sentences are of particular interest because they are clear and concise biographical summaries. These could be applied to entities outside Wikipedia for which one can obtain comparable parallel structured/textual data, e.g., movie summaries from IMDb, resume overviews from LinkedIn, product descriptions from Amazon.

We use snapshots of Wikidata (2015/07/13) and Wikipedia (2015/10/02) and batch process them to extract instances for learning. We select all entities that are `INSTANCE_OF human` in Wikidata. We then use `sitelinks` to identify each entity's

Fact	Count	%
TITLE (name)	1,011,682	98
SEX_OR_GENDER	1,007,575	0
DATE_OF_BIRTH	817,942	88
OCCUPATION	720,080	67
CITIZENSHIP	663,707	52
DATE_OF_DEATH	346,168	86
PLACE_OF_BIRTH	298,374	25
EDUCATED_AT	141,334	32
SPORTS_TEAM	108,222	29
PLACE_OF_DEATH	107,188	17
POSITION_HELD	87,656	75
PARICIPANT_OF	77,795	23
POLITICAL_PARTY	74,371	49
AWARD_RECEIVED	67,930	44
SPORT	36,950	72

Table 6.1: The top fifteen slots across entities used for input, and the % of time the value is a substring in the entity’s first sentence.

Wikipedia article text and NLTK (Bird et al., 2009) to tokenize and extract the lower-cased first sentence. This results in 1,268,515 raw knowledge-text pairs. The summary sentences can be long and the most frequent length is 21 tokens. We filter to only include those between the 10th and 90th percentiles: 10 and 37 tokens. We split this collection into TRAIN, DEV and TEST collections with 80%, 10% and 10% of instances allocated respectively. Given the large variety of slots which may exist for an entity, we restrict the set of slots used to the top-15 by occurrence frequency. This criteria covers 72.8% of all facts. Table 6.1 shows the distribution of fact slots in the structured data and the percentage of time tokens from a fact value occur in the corresponding Wikipedia summary.

Additionally, some Wikidata entities remain underpopulated and do not contain sufficient facts to reconstruct a text summary. We control for this information mismatch

by limiting our dataset to include only instances with at least 6 facts present. The final dataset includes 401,742 TRAIN, 50,017 DEV and 50,030 TEST instances. Of these instances, 95% contain 6 to 8 slot values while 0.1% contain the maximum of 10 slots. 51% of unique slot-value pairs expressed in TEST and DEV are not observed in TRAIN so generalisation of slot usage is required for the task. The KB facts give us an opportunity to measure the correctness of the generated text in a more precise way than text-to-text tasks. We use this for analysis in Section ??, driving insight into system characteristics and implications for use.

6.3.1 Task complexity

Wikipedia first sentences exhibit a relatively narrow domain of language in comparison to other generation tasks such as translation. As such, it is not clear how complex the generation task is, and we first try to use perplexity to describe this.

We train both RNN models until DEV perplexity stops improving. Our basic sequence-to-sequence model (S2S) reaches perplexity of 2.82 on TRAIN and 2.92 on DEV after 15,000 batches of stochastic gradient descent. The autoencoding sequence-to-sequence model (S2S+AE) takes longer to fit, but reaches a lower minimum perplexity of 2.39 on TRAIN and 2.51 on DEV after 25,000 batches.

To help ground perplexity numbers and understand the complexity of sentence biographies we train a benchmark language model and evaluate perplexity on DEV. Following Lebre et al. (2016), we build Kneser-Ney smoothed 5-gram language models using the KenLM toolkit (Heafield, 2011).

Table 6.2 lists perplexity numbers for the benchmark LM models with different templating schemes on DEV. We observe decreasing perplexity for data with greater fact value templating. TITLE indicates templating of entity names only, while FULL indicates templating of all fact values by token index as described in Lebre et al. (2016). This shows that templating is an effective way to reduce the sparsity of a task, and that titles account for a large component of this.

Templates	DEV
None	29.8
Title	14.5
Full	10.1

Table 6.2: Language model perplexity across templated datasets.

Although Lebre et al. (2016) evaluate on a different dataset, we are able to draw some comparisons given the similarity of our task. On their data, the benchmark LM baseline achieves a similar perplexity of 10.5 to ours when following their templating scheme on our dataset - suggesting both samples are of comparable complexity.

6.4 Model

We model the task as a sequence-to-sequence learning problem. In this setting, a variable length input sequence of entity facts is encoded by a multi-layer RNN into a fixed-length distributed representation. This input representation is then fed into a separate decoder network which estimates a distribution over tokens as output. During training, parameters for both the encoder and decoder networks are optimized to maximize the likelihood of a summary sequence given an observed fact sequence.

Our setting differs from the translation task in that the input is a sequence representation of structured data rather than natural human language. As described above in Section 6.3, we map Wikidata facts to a sequence of tokens that serves as input to the model as illustrated at the top of Figure 6.2. Experiments below demonstrate that this is sufficient for end-to-end learning in the generation task addressed here. To generate summaries, our model must both select relevant content and transform it into a well formed sentence. The decoder network includes an attention mechanism (Vinyals et al., 2015b) to help facilitate accurate content selection. This allows the network to focus on different parts of the input sequence during inference.

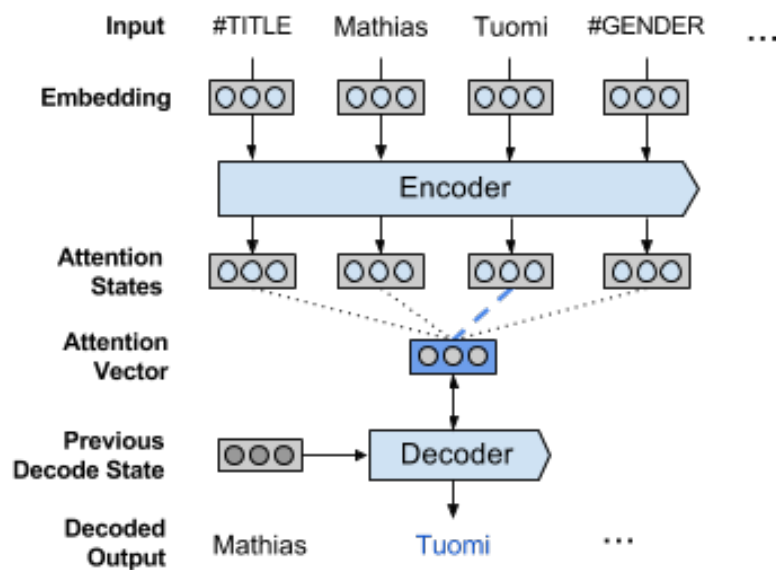


Figure 6.2: Sequence-to-sequence translation from linearized facts to text.

6.4.1 Sequence-to-sequence model (S2S)

To generate language, we seed the decoder network with the output of the encoder and a designated GO token. We then generate symbols greedily, taking the most likely output token from the decoder at each step given the preceding sequence until an EOS token is produced. This approach follows (Sutskever et al., 2014) who demonstrate a larger model with greedy sequence inference performs comparably to beam search. In contrast to translation, we might expect good performance on the summarization task where output summary sequences tend to be well structured and often formulaic. Additionally, we expect a partially-shared language across input and output. To exploit this, we use a tied embedding space, which allows both the encoder and decoder networks to share information about word meaning between fact values and output tokens.

Our model uses a 3-layer stacked Gated Recurrent Unit RNN for both encoding and decoding, implemented using TensorFlow.¹ We limit the shared vocabulary to 100,000 tokens with 256 dimensions for each token embedding and hidden layer. Less common

¹<https://www.tensorflow.org>, v0.8.

tokens are marked as UNK, or unknown. To account for the long tail of entity names, we replace matches of title tokens with templated copy actions (e.g. TITLE0 TITLE1 . . .). These template are then filled after generation, as well as any initial unknown tokens in the output, which we fill with the first title token. We learn using minibatch Stochastic Gradient Descent with a batch size of 64 and a fixed learning rate of 0.5.

6.4.2 S2S with autoencoding (S2S+AE)

One challenge for vanilla sequence-to-sequence models in this setting is the lack of a mechanism for constraining output sequences to only express those facts present in the data. Given a fact extraction oracle, we might compare facts expressed in the output sequence with those of the input and appropriately adjust the loss for each instance. While a forward-only model is only constrained to generate text sequences predicted by the facts, an autoencoding model is additionally constrained to generate text predictive of the input facts.

In place of this ideal setting, we introduce a second sequence-to-sequence model which runs in reverse — re-encoding the text output sequence of the forward model into facts. For an input set of facts x and target output sequence y we construct the forward S2S model F_{fwd} as normal and predict an output sequence y' .

$$y' = F_{fwd}(x)$$

We then feed the output of this forward network as input into a second S2S model F_{bwd} with the input x as the target prediction sequence x' .

$$x' = F_{bwd}(y')$$

As before we share embedding parameters between the source and target vocabulary for the forward model and additionally share these parameters as the source and target vocabulary for the backwards model. All other model parameters are decoupled. This closed-loop model is detailed in Figure 6.3. The resulting network is trained end-to-end

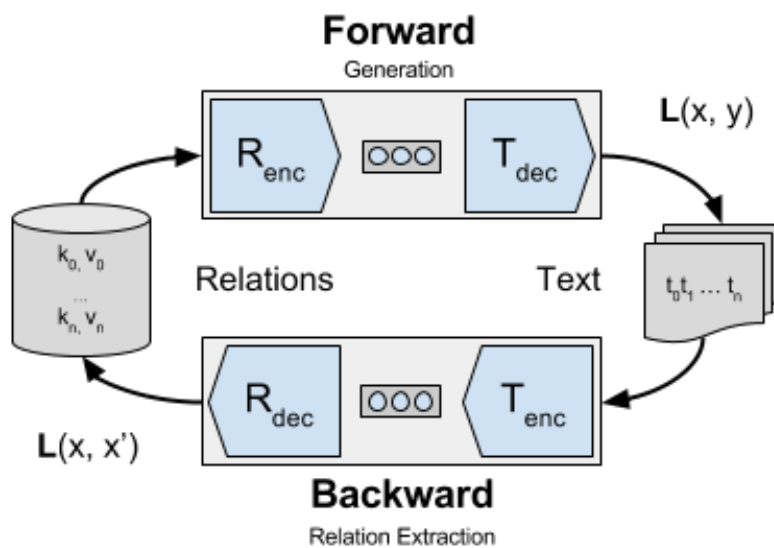


Figure 6.3: Sequence-to-sequence autoencoder.

to minimize the input-to-output sequence loss $L_{fwd}(y, y')$ of the forward model and output-to-input sequence reconstruction loss $L_{bwd}(x, x')$ of the backward model with equal weight.

Under this network architecture gradients cannot propagate back through the greedy forward sequence decoding step, however the combined model can benefit from shared parameters fit on the multi-task encode-decode objective. To generate text at test time, we need not evaluate the backward network – we revisit the idea of decoding fact sequences from text in Chapter 7.

6.5 Experimental methodology

The evaluation suite here includes standard baselines for comparison, automated metrics for learning, human judgement for evaluation and detailed analysis for diagnostics. While each are individually useful, their combination gives a comprehensive analysis of a complex problem space.

6.5.1 Benchmarks

WIKI We use the first sentence from Wikipedia both as a gold standard reference for evaluating generated sentences, and as an upper bound in human preference evaluation.

BASE Template-based systems are strong baselines, especially in human evaluation. While output may be stilted, the corresponding consistency can be an asset when consistency is important. We induce common patterns from the TRAIN set, replacing full matches of values with their slot and choosing randomly on ties. Multiple non-fact tokens are collapsed to a single symbol. A small sample of the most frequent patterns were manually examined to produce templates, roughly expressed as: "TITLE, known as GIVEN_NAME, (born DATE_OF_BIRTH in PLACE_OF_BIRTH; died DATE_OF_DEATH in PLACE_OF_DEATH) is an POSITION_HELD and OCCUPATION from CITIZENSHIP", with some sensible back-offs where slots are not present, and rules for determiner agreement and is versus was where a death date is present. For example, "ollie freckingham (born 12 november 1988) is a cricketer from the united kingdom". In total, there are 48 possible template variations.

6.5.2 Metrics

6.5.2.1 BLEU

We also report BLEU n-gram overlap with respect to the reference Wikipedia summary. With a large dev/test sets (10,000 sentences here), BLEU is a reasonable evaluation of generated content. However, it does not give an indication of well-formedness or readability. Thus we complement BLEU with a human preference evaluation.

6.5.2.2 Human preference

We use crowd-sourced judgements to evaluate the relative quality of generated sentences and the reference Wikipedia first sentence. We obtain pairwise judgements, showing output from two different systems to crowd workers and ask each to give their binary preference. The system name mappings are anonymized and ordered pseudo-randomly. We do not provide the reference facts for a summary and simply ask annotators: "Do you prefer summary A to summary B" — as such, we expect annotations to primarily measure the interpretability or fluency of a summary in place of its factual correctness. We request 3 judgements and dynamically increase this until we reach at least 70% agreement or a maximum of 5 judgements. We use CrowdFlower² to collect judgements at the cost of 31 USD for all 6 pairwise combinations over 82 randomly selected entities. 67 workers contributed judgements to the test data task, each providing no more than 50 responses. We use the majority preference for each comparison. The CrowdFlower agreement is 80.7%, indicating that roughly 4 of 5 votes agree on average.

6.5.3 Analysis of content selection

Finally, no system is perfect, and it can be challenging to understand the inherent difficulty of the problem space and the limitations of a system. Due to the limitations of the evaluation metrics mentioned above, we propose that manual annotation is important and still required for qualitative analysis to guide system improvement. The structured data in knowledge-to-text tasks allows us, if we can identify expressions of facts in text, cases where facts have been omitted, incorrectly mentioned, or expressed differently.

²<http://www.crowdfunder.com>

	DEV	TEST
Base	21.3	21.1
S2S	32.5	33.1
S2S+AE	40.5	41.0

Table 6.3: BLEU scores for each hypothesis against the Wikipedia reference

6.6 Results

6.6.1 Comparison against Wikipedia reference

Table 6.3 shows BLEU scores calculated over 10,000 entities sampled from DEV and TEST using the Wikipedia sentence as a single reference, using uniform weights for 1- to 4-grams, and padding sentences with fewer than 4 tokens. Scores are similar across DEV and TEST, indicating that the samples are of comparable difficulty. We evaluate significance using bootstrapped resampling with 1,000 samples. Each system result lies outside the 95% confidence intervals of other systems. BASE has reasonable scores at 21, with S2S higher at around 32, indicating that the model is at least able to generate closer text than the baseline. S2S+AE scores higher still at around 41, roughly double the baseline scores, indicating that the autoencoder is indeed able to constrain the model to generate better text.

6.6.2 Human preference evaluation

Table 6.4 shows the results of our human evaluation over 82 entities sampled from TEST. For each pair of systems, we show the percentage of entities where the crowd preferred A over B. Significant differences are annotated with * and ** for p values < 0.05 and 0.01 using a one-way χ^2 test. WIKI is uniformly preferred to any system, as is appropriate for an upper bound. The S2S model is the least-preferred with respect to WIKI. The S2S+AE model is more-preferred than the BASE and S2S models, by a larger

S2S+AE	BASE	S2S	
60%	61%*	87%**	WIKI
	62%*	77%**	S2S+AE
		65%**	BASE

Table 6.4: Percentage of entities for which human judges preferred the row system to the column system. E.g., S2S+AE summaries are preferred to BASE for 62% of sample entities.

margin for the latter. These results show that without autoencoding, the sequence-to-sequence model is less effective than a template-based system. Finally, although WIKI is more preferred than S2S+AE, the distributions are not significantly different, which we interpret as evidence that the model is able to generate good text from the human point-of-view, but autoencoding is required to do so.

6.7 Analysis

While results presented above are encouraging and suggest that the model is performing well, they are not diagnostic in the sense that they can drive deeper insights into model strengths and weaknesses. While inspection and manual analysis is still required, we also leverage the structured factual data inherent to our task to perform quantitative as well as qualitative analysis.

6.7.1 Fact Count

Figure 6.4 shows the effects of an increasing input fact count on generation performance as measured by BLEU score. While more input facts give more information for the model to work with, longer inputs are also both rarer and more complex to encode. Interestingly, we observe the S2S+AE model maintains performance for more complex inputs while S2S performance declines.

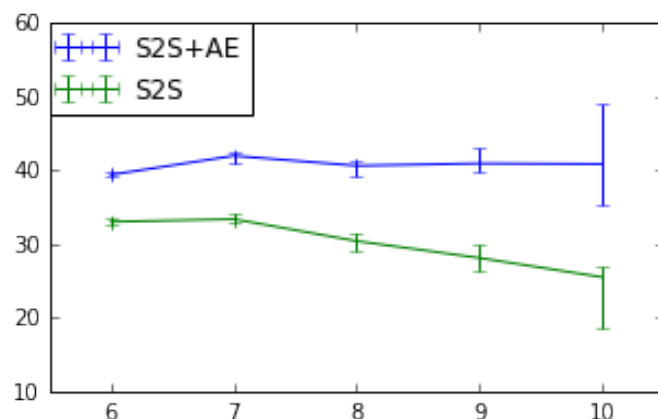


Figure 6.4: BLEU vs Fact Count on instances from DEV. Error bars indicate the 95% confidence interval for BLEU.

6.7.2 Example generated text

Table 6.5 shows some DEV entities and their summaries. The model learns interesting mappings: between numeric and string dates, and country demonyms. The model also demonstrates the ability to work around edge cases where templates fail, i.e. stripping parenthetical disambiguations (e.g. (actor)) and emitting the name Robert when the input is Bob. Output also suggests the model may perform inference across multiple facts to improve generation precision, e.g. describing an entity as english rather than british given information about both citizenship and place of birth. Unfortunately, the model can also infer unsubstantiated facts into the text (i.e. jazz drummer).

Data		COUNTRY_OF_CITIZENSHIP united states of america DATE_OF_BIRTH 16/04/1927 DATE_OF_DEATH 19/05/1959 OCCUPATION formula one driver PLACE_OF_BIRTH redlands PLACE_OF_DEATH indianapolis SEX_OR_GENDER male TITLE bob cortner
WIKI	n/a	robert <i>charles</i> cortner (april 16 , 1927 - may 19 , 1959) was an american automobile racing driver from <i>redlands , california</i> .
BASE	47.7	bob cortner (born 16 april 1927 in redlands ; died 19 may 1959 in indianapolis) was a formula one driver from the united states of america
S2S	45.7	bob cortner (april 16 , 1927 - may 19 , 2005) was an american professional boxer .
S2S+AE	58.8	robert cortner (april 16 , 1927 - may 19 , 1959) was an american racecar driver .
Data		COUNTRY_OF_CITIZENSHIP united kingdom DATE_OF_BIRTH 08/01/1906 DATE_OF_DEATH 12/12/1985 OCCUPATION actor PLACE_OF_BIRTH london PLACE_OF_DEATH chelsea SEX_OR_GENDER male TITLE barry mackay (actor)
WIKI	n/a	barry mackay (8 january 1906 - 12 december 1985) was a british actor.
BASE	34.3	barry mackay (actor) (born 8 january 1906 in london ; died 12 december 1985 in chelsea) was an actor from the united kingdom .
S2S	84.8	barry mackay (8 january 1906 - 12 december 1985) was a british film actor .
S2S+AE	76.7	barry mackay (8 january 1906 - 12 december 1985) was an english actor .
Data		COUNTRY_OF_CITIZENSHIP united states of america DATE_OF_BIRTH 27/08/1931 DATE_OF_DEATH 03/11/1995 OCCUPATION jazz musician SEX_OR_GENDER male TITLE joseph "flip" nuñez
WIKI	n/a	joseph “ <i>flip</i> ’ nuñez was an american jazz pianist , composer , and vocalist of <i>filipino</i> descent .
BASE	15.0	joseph “ <i>flip</i> ’ nuñez (born 27 august 1931 ; died 3 november 1995) was a jazz musician from the united states of america .
S2S	29.1	joseph “ <i>flip</i> ’ nuñez (august 27 , 1931 - november 3 , 1995) was an american jazz trumpeter .
S2S+AE	29.1	joseph “ <i>flip</i> ’ nuñez (august 27 , 1931 - november 3 , 1995) was an american jazz drummer .

Table 6.5: Input facts and output summaries for each system over entities sampled from DEV. We mark **correct**, **incorrect** and *extra* fact values in the text with respect to the Wikidata input.

6.7.3 Content selection and hallucination

We randomly sample 50 entities from DEV and manually annotate the Wikipedia and system text. We note which fact slots are expressed as well as whether the expressed values are correct with respect to Wikidata. Given two sets of correctly extracted facts, we can consider one *gold*, one *system* and calculate set-based precision, recall and F1.

6.7.3.1 What percentage of facts are used in the reference summaries?

Firstly, to understand how Wikipedia editors select content for the first sentence of articles, we measure recall with the real facts as gold, and Wikipedia as system. Overall, the recall is 0.61 indicating that 61% of input facts are expressed in the reference summary from Wikipedia. The entity name (TITLE) is always expressed. Four slots are nearly always expressed when available: OCCUPATION (90%), DATE_OF_BIRTH (84%), CITIZENSHIP (81%), DATE_OF_DEATH (80%). Six slots are infrequently expressed in the analysis sample: PLACE_OF_BIRTH (33%), POSITION_HELD (25%), PARTICIPANT_OF (20%), POLITICAL_PARTY (20%), EDUCATED_AT (14%), SPORTS_TEAM (9%). Two are never expressed explicitly: PLACE_OF_DEATH (0%), SEX_OR_GENDER (0%). AWARD_RECEIVED and SPORT are not in the analysis sample.

6.7.3.2 Do systems select the same facts found in the reference summaries?

Table 6.6 shows content selection scores for systems with respect to the Wikipedia text as reference. This suggests that the autoencoding in S2S+AE helps increase fact recall without sacrificing precision. The template baseline also attains this higher recall, but at the cost of precision. For commonly expressed facts found in most person biographies, recall is over 0.95 (e.g., CITIZENSHIP, BIRTH_DATE, DEATH_DATE and OCCUPATION). Facts that are infrequently expressed are more difficult to select, with system F1 ranging from 0.00 to 0.50. Interestingly, macro-averaged F1 across infrequently expressed facts mirror human preference rather than BLEU results, with

	P	R	F
BASE	0.80	0.79	0.79
S2S	0.89	0.67	0.77
S2S+AE	0.89	0.78	0.83

Table 6.6: Fact-set content selection results phrased as precision, recall and F1 of systems with respect to the Wikipedia reference on DEV.

System	Mean facts	Mean tokens
BASE	5.1	21.2
S2S	4.6	19.7
S2S+AE	5.2	19.1
WIKI	6.1	23.7

Table 6.7: Fact density and sentence length analysis.

S2S+AE (0.26) > BASE (0.17) > S2S (0.07). However, all systems perform poorly on these facts and no reliable differences are observed.

6.7.3.3 How does autoencoding effect fact density?

Interestingly, we observe that the autoencoding objective encourages the model to select more facts (5.2 for S2S+AE vs. 4.5 for S2S), without increasing sentence length (19.1 vs. 19.7 tokens). BASE is similarly productive (5.1 facts) but wordier (21.2 tokens), while the WIKI reference produces both more facts (6.1) and longer sentences (23.7).

Table 6.7 shows average numbers of tokens and facts found in the different outputs. In general, Wikipedia sentences are the longest and contain the most information. The baseline contains a similar amount of data to S2S+AE, but uses more tokens. S2S sentences are longer on average than S2S+AE despite containing less facts. This suggests our autoencoding model is better able to concisely convey information.

	P	R	F
BASE	1.00	0.74	0.85
S2S	0.96	0.55	0.70
S2S+AE	0.93	0.62	0.74
WIKI	0.81	0.61	0.69

Table 6.8: Hallucination results phrased as precision, recall and F1 of systems with respect to the Wikidata input on DEV.

6.7.3.4 Do systems hallucinate facts?

To quantify the effect of hallucinated facts, we assess content selection scores of systems with respect to the input Wikidata facts (Table 6.8). Our best model achieves a precision of 0.93 with respect to Wikidata input. Notably, the template-driven baseline maintains a precision of 1.0 as it is constrained to emit Wikidata facts verbatim.

6.8 Discussion

Evaluation in this domain is challenging. In place of a single score, we analyse statistical measures, human preference judgements and manual annotation to help characterize the task and understand system performance.

Our text generation model is able to replicate the Wikipedia biographic style, outperforming template baselines and achieving preference over reference summaries in 40% of cases evaluated by human judges. However, we also observe a tendency for the model to express facts about an entity which may be unfounded given the inputs. In applications where the precision of generated text is paramount, template driven approaches are still preferable, despite trade-offs in readability and conciseness. Hybrid approaches which dynamically generate and copy text (Vinyals et al., 2015a; Gu et al., 2016; Jia and Liang, 2016) from the input present an interesting compromise

between templatised and model-driven NLG. Copy actions may also help to mitigate for vocabulary constraints and better generalise to unseen entities and relations.

Our system is able to model the likelihood of language conditioned on established facts from a KB. This framework may be useful in other translation applications — for example, by integrating structured knowledge into traditional translation models, we may better translate descriptions from KB entries for well populated languages to those with less coverage. To address multi-sentence generation, we expect conditioning of a simple single sentence model on the target sentence index would be sufficient to model generation for simple or well structured target language domains (i.e. introducing $BOS_0, BOS_1 \dots BOS_k$ tokens). For more complex applications, conditioning via extended model parameterisation and the passing of state vectors between sentence generators is likely necessary to better model long-form text generation. Text generation aside, we may also find applications for our model in KB consistency checking. If a given KB edit is judged unlikely under a set of facts for an entity, this may suggest that either the change is nefarious or the facts are wrong — in either case, flagging the entry for review may aid KB curation.

Similar RNN models have been applied extensively to language translation tasks. Joint model of machine translation and fact-driven generation may help populate KB entries for low-coverage languages from a shared set of facts. Our analysis shows that robust fact-based summary evaluation is challenging. While work in distributional semantics may help derive better metrics (Passonneau et al., 2013), we might also incorporate relation extraction to automatically assess content selection and realisation quality.

6.9 Summary

In this chapter we address the task of biography generation. We develop a neural network translation model which encodes linearized facts and decodes one-sentence

entity biographies. We expect this model to find applications in both the curation and population of entity descriptions within a KB. While we observe that a secondary autoencoding objective is able to improve the quality of generated text, we have not yet attempted to assess the performance of translation over the reverse task of generating facts from text.

This task is a critical precursor for description generation in web KBs where no structured entity representation yet exists. In the next chapter we describe this task in detail and present a preliminary adaptation of the knowledge translation framework to fact generation from entity references in text.

7 Fact inference

Everything should be made as simple
as possible, but no simpler.

Einstein

In the previous chapter we take as given a set of facts describing an entity and aim to generate a textual summary description. In this and many other tasks we value a structured representation of entity knowledge. Our work has so far only considered the aggregation of unstructured knowledge from web — i.e. natural language mentions of an entity annotated via web KB links. For this data to be useful in downstream applications, a categorization and canonicalization of latent textual knowledge is often desirable.

This chapter explores the task of **fact inference**. Given one or more mentions of an entity in text, we aim to predict well-formed facts under a fixed KB schema. Here again we build upon the framework of knowledge translation. In place of generating text from facts, we attempt to generate fact values conditioned on textual descriptions of an entity. Our model is able to both learn to generate canonicalized fact values under the target KB schema and infer the value of facts which are never made explicit in text.

Our experiments consider inference over both entity biographies (i.e. mirroring the setup of text generation experiments) and inbound links to an entity page (i.e. simulating the setting of web KB construction). In combination with experiments in Chapter 6, our models provide a mechanism for both distilling structured knowledge from mentions and concisely describing an entity from inferred facts.

Our contributions include: (1) a multi-output sequence-to-sequence translation model for fact inference; (2) detailed analysis of fact inference models over biographic summaries and linked entity mentions; (3) annotated data for analysis of fact explicitness in text; This chapter describes preliminary work which has not previously been published under peer review. Code and annotation from our experiments are available at: github.com/andychisholm/mimo .

7.1 Introduction

Traditional KBs like Wikidata and Freebase maintain a curated schema of structured facts for each entity. These facts are often expressible as relational triples encoding an entity subject, relational predicate and object value. For example, we may encode the knowledge that "Elon Musk is the CEO of Tesla" in the triple: (Elon Musk, CEO, Tesla Inc.). This encoding simplifies question answering by enabling questions to be encoded as queries over structured facts. For example, "Who is the CEO of Tesla" becomes a search for triples of the form (?, CEO, Tesla Inc.). Facts represent a canonical encoding of knowledge about an entity. While there may be many ways to express a fact in text, there typically exists just one canonical encoding under a given KB schema.

We investigate the task of fact inference over entity mentions in text. Our formulation of this task is guided by the setting of web KB construction. Given a corpus of textual mentions resolved to specific KB entities via web links, we wish to infer a structured knowledge representation for use in downstream tasks such as question answering and structured search. Moreover, we aim to further explore the knowledge translation framework described for text generation experiments in Chapter 6.

We develop a fact inference model which addresses the previously considered input-output transformation in reverse — in-place of generating text conditioned on entity facts, we generate fact values conditioned on the text surrounding entity mentions. Building on a base model from machine translation, we adapt our approach to better

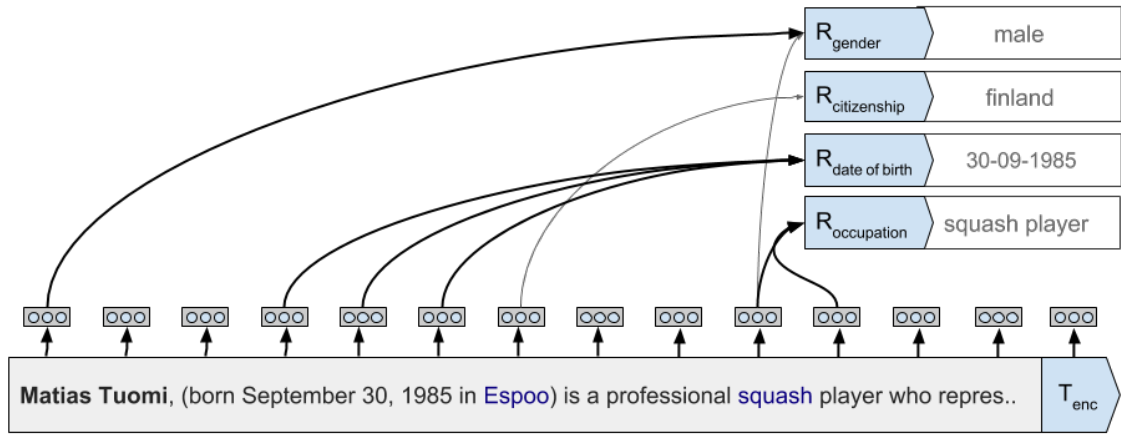


Figure 7.1: Multi-fact inference over a shared input representation. Arrows indicate selective attention over the input by each decoder.

address the structure of the fact inference task. As a single input sentence may be predicative of multiple fact values, we directly model this one-to-many relationship by sharing a common text encoder network across multiple independent decoders for each fact type. Each fact decoder learns to attend the input for information relevant to a specific fact and emit values within a closed vocabulary relevant to that type. We detail this adapted translation model in Figure 7.1. In contrast with extraction driven approaches to KB population, end-to-end translation breaks the tight coupling between the surface form of information in text and structured facts to be populated in the KB. For example, we may leverage information from a given name to predict gender or use a stated place of birth to predict citizenship. Translation also implicitly addresses transformations of information into the target schema, e.g. conversion of a written date "September 30, 1989" into the numerical equivalent "30-09-1989". Where extraction driven systems require higher-order methods to address issues of schema mapping and inference, translation provides a direct mechanism for resolving these tasks locally over text.

We compare two distinct configurations of the fact inference task on a recent sample of entity facts and corresponding mentions extracted from Wikipedia and Wikidata. In the first case we explore fact inference over the encyclopedic summary of a target

entity — mirroring the setup of text generation experiments and shedding light on how fact inference models may leverage existing descriptions to populate missing information within a traditional KB. Next, we seek to simulate the web-KB construction setting by inferring facts from inbound links to an entity page — without utilizing information from the entity page itself. In each case, our model is able to decode important facts about entity identity such as gender, occupation and citizenship with high precision (50-95% at R=1). In aggregate we observe significantly higher performance across fact types extracting information from the biographic summary, especially for facts which are rarely made explicit outside biographical descriptions in text (e.g. date of birth). To better understand fact expression around inbound links to an entity we manually analyze the explicitness of expressed facts across a subset of fact types, observing that while most facts are rarely made explicit, the reference value for a fact may be recovered in the majority of cases by reasoning over the expressed information and priors from the KBs. Our models and analysis complement existing work on structured KB population and further develop the knowledge translation framework.

7.2 Related work

Neural networks have become an increasingly common feature of high-performance information extraction systems. The integration of convolutional models (Zeng et al., 2014; Nguyen and Grishman, 2015), recurrent neural networks (Cai et al., 2016) and neural attention (Zhou et al., 2016; Lin et al., 2016; Verga et al., 2018) have steadily increased performance in both sentence level relation extraction (RE) and end-to-end KBP tasks such as Slot Filling (SF) (Adel et al., 2016; Huang et al., 2017). In this section we focus on connections between our specific formulation of the fact interface task and existing approaches to KB population and question answering.

Wu and Weld (2007) present one of the first such approaches to structured information extraction with Wikipedia. They extract info-box tables from article text by tagging token sequences which denote attribute values for an entity. To train their system, they sample instances where populated info-box values appear in the corresponding article text — leveraging existing knowledge to heuristically label instances of attribute expression. Mintz et al. (2009) generalize this approach, introducing the distant supervision framework for relation extraction (RE) whereby any sentence which mentions a pair of related entities is assumed to express that relation in text. A substantial body of work builds upon this heuristic by relaxing the assumption of one-to-one alignment between mentioned entities and expressed facts. Riedel et al. (2010) demonstrate an expressed-at-least-once assumption across mentioned entity pairs reduces the impact of distant supervision noise and Surdeanu et al. (2012) propose a multi-instance multi-label learning framework which additionally models instances of multiple relations between mentioned entity pairs. Our formulation of the fact inference task represents a continuation of this trend towards relaxing the distant supervision assumption. Rather than constraining training instances to sentences where a pair of related entities are mentioned, we allow any mention of a subject entity to be predictive of any value across populated predicates for that entity in the KB. Under this training objective, our model must distill salient source information to predict fact values which may never be expressed in the input.

Extraction driven approaches to structured KB population are well suited to relational facts where both the subject and object of a predicate are named entities which appear in text. However, many interesting facts we may wish to infer about an entity may not be explicitly described. For example, given a sentence such as "John was born in San Francisco, California", we should be able to reason about the likelihood of citizenship within the United States with high-confidence. While we may hope to find some other specific textual reference to facts of this type, a wide range of common-sense knowledge is rarely made explicit in text (Liu and Singh, 2004; Angeli

and Manning, 2013). For example, if we never observe a sentence such as "John is a man", we may never fill the gender slot. In principle, we ought to be able to infer this fact from references which don't explicitly mention it, i.e. via knowledge that John is a common male name. Knowledge Base Completion (KBC) systems (Singh and Gordon, 2008; Bordes et al., 2011; García-Durán et al., 2016) address this problem in part by attempting to infer unknown relations about an entity from those which are present in the KB. For example, if we are able to extract (John, place_of_birth, San Francisco) and have prior knowledge that (San Francisco, country, United States) we may subsequently infer the fact (John, citizenship, United States). In addition, they present a mechanism whereby OpenIE systems (Banko et al., 2007; Mausam et al., 2012) which extract redundant relational forms in terms of the source language itself may be mapped onto a fixed and finite relational schema (Riedel et al., 2013). State-of-the-art KBC systems predict unseen relations between candidate entities by jointly embedding entities and relations within a latent feature space, then classify the likelihood of new relations between candidate triples (Nguyen, 2017). These KB level representations may be also integrated within a RE model to further improve performance (Weston et al., 2013; Riedel et al., 2013; Toutanova et al., 2015). By contrast, our fact inference model aims to resolve implicit expressions of fact over text alone. This approach does not preclude downstream applications of KBC, but rather increases the number of facts available to higher-order systems via low-level inference over non-relational textual associations — e.g. leveraging gendered pronouns (i.e. she, her, he, him) to predict gender or learning associations between expressions of residence and citizenship (i.e. Californians often hold United States citizenship).

Systems which attend textual input to generate answers for questions (Weston et al., 2015) or structured responses (Palm et al., 2017) via sequence to sequence modelling are also closely related. As are semantic parsing (Woods, 1973) systems which transform text (e.g. questions from a natural language dialog) into an equivalent formal meaning representation via sequence-to-sequence models (e.g. Dong and Lapata (2016); Duong

et al. (2018)). In this domain, Hewlett et al. (2016) present the closest work to our own with the WikiReading task. Building on the info-box attribute extraction task of Wu and Weld (2007), they aim to extract Wikidata facts from an entity’s Wikipedia article. In comparison to our work, they source a larger sample of page content (up-to 300 words) and target a more diverse set of 884 fact types. They also investigate a large variety of alternative models including bag-of-words classifiers, extractive methods, memory networks (Sukhbaatar et al., 2015) and sequence-to-sequence generation models. By contrast, we focus on sentence level context and additionally consider inference over inbound links to an entity page — a source which our analysis suggests is both less information dense and more loosely structured than the canonical encyclopedic entry for an entity.

7.3 Data

Despite the similarity between each task, our construction of a dataset for text generation experiments in 6.3 targets biographies alone — without the inclusion of text sequences from inbound links to an entity page. Moreover, constraints imposed on the inclusion of instances by the number of facts present need not apply to the inverse task of fact inference.

To explore the fact inference task, we extract textual entity mentions and corresponding entity facts from updated snapshots of both Wikipedia (2017/03/01) and Wikidata (2017/03/06); normalizing fact values (e.g. dates) in an equivalent manner to that described in 6.3. For symmetry with text generation experiments, we once again select entities from instances of the *humans* type and target the same subset of Wikidata relations. Over a total of approximately 4M entities with Wikipedia alignment we extract a total of 1.4M *humans* entity instances from Wikidata. We split instances into discrete TRAIN, DEV and TEST collections with 80%, 10% and 10% of entity instances allocated respectively.

In general we have no way of knowing whether information in the source text for an entity parallels that of the facts present. For experiments in this section, we impose no constraint on the number of mentions or relations present for an entity and instead attempt to quantify the impact of input-output information disparity as part of our analysis in Section 7.6.

7.3.1 Facts

Table 7.1 lists statistics of fact values by type for the updated human entity corpus. For each fact type we denote: Occurrence (%) - the percentage of instances for which the relation is present; Vocab - the number of unique fact values; Most Common Value - the value string with the highest frequency across all relation values; and Coverage - the percentage of instances populated with the most common value.

While there is clearly great disparity in the types of relations present across human entities in Wikidata, some of the most important information in characterizing basic entity identity are highly populated, i.e. `gender`, `occupation` and `date of birth`. There is also clearly great disparity in the size of the vocabulary needed to describe each relation. While only 11 values describe the variety of gender types encoded in Wikidata, fields with named entity fills (e.g. `place of death`) have far more variation. Value coverage is also interesting to consider. For facts like `gender` and `sport` the most common values of `male` and `association football` respectively account for the majority of the value frequency distribution. In the case of near-fully populated facts such as `gender`, this skew in the natural distribution indicates a bias toward coverage for entities of that type. For facts with lower occurrence rates, skews in coverage for certain values may indicate integration of knowledge resources from other domain specific KBs into Wikidata, or simply preferential curation of entities in notable categories, e.g. `football players` and `Harvard graduates`.

Fact Type	Occ. (%)	Vocab	Most Common Value	Cov. (%)
sex or gender	99.7	11	male	83.5
date of birth	84.9	107,940	2000 01 01	0.2
occupation	79.8	2,993	politician	11.7
given name	79.0	16,991	john	3.3
citizenship	73.3	1,059	united states of america	27.7
place of birth	58.2	78,775	new york city	1.5
date of death	39.6	107,675	2000 01 01	0.1
place of death	19.6	34,722	paris	3.8
educated at	18.3	10,772	harvard university	3.8
sport	16.2	249	association football	57.2
sports team	15.3	15,301	st . louis cardinals	0.5
position held	9.9	5,144	united states representative	6.2
award received	8.5	5,257	guggenheim fellowship	4.7
family name	8.2	13,001	smith	4.3
participant of	7.8	4,550	2008 summer olympics	6.5
political party	7.5	3,069	democratic party	18.7

Table 7.1: Percentage of the frequency distribution for the most common value of each Wikidata fact. *Occ.* denotes the percentage of instances for which the fact is populated. *Vocab* denotes the number of unique values. *Cov.* denotes the percentage of instances for which the most common value for that slot is the slot value.

7.3.2 Text

Experiments in Section 6.1 target first sentence summaries from an entity Wikipedia article. While these sentences present an ideal target for fact inference, they are clearly not representative of general entities references across the web. In this section we aim to evaluate fact inference for both biographic summaries and the more general case of entity mentions in text.

In general we expect web mentions to be both less fact-dense and more linguistically complex than the consistently formed summaries found in Wikipedia (cf. our analysis

LHS	Span	RHS
... of billionaire businessman	elon musk	and a major tesla shareholder ...
... 2016 , tesla motors ceo	elon musk	stated that apple will probably ...
... wave of grants in which	elon musk	participated .
... a tour of spacex by	elon musk	.
... meeting between calacanis and	elon musk	, musk mentioned that the ...
... was given by spacex ceo	elon musk
... partner presented the startup to	elon musk	during the under 30 summit ...
... march 2016 , tesla ceo	elon musk	announced that the number of ...
...	elon musk	was the film 's executive ...
...	elon musk	on a march 2015 tour ...
... hoffman , peter thiel ,	elon musk	, ben horowitz and tony ...
... the neurosciences institute , and	elon musk	, co-founder of paypal , ...
... one of	elon musk	's stated goals through his ...
... , associated with business magnate	elon musk	, that aims to carefully ...
... vision of spacex , ceo	elon musk	, to begin colonizing mars ...

Table 7.2: Random sample of 15 inlinks for the Elon Musk Wikipedia article. While sentences are truncated for display here the generated dataset includes full sentence spans for each inlink. No linked mentions appear within the entity article itself.

in Section 6.3.1). To better approximate the setting of fact inference over inbound links to an entity, we extract sentences enclosing links to an entity page across Wikipedia. Over our collection of 1.4M human entities we extract a total of 13.2m entity links — an average of 9.3 mentions per entity. Table 7.2 shows a random sample of mentions for the Elon Musk entity. For each inlink, we extract the surrounding sentence context and record the position of the link anchor span within the sentence. We limit the context of inlinks to a single enclosing sentence for simplicity alone. While we expect greater document context and coreferential mentions around each inlink to be a rich source of entity knowledge, we leave the incorporation of deeper document context for future work.

Fact Type	Value
sex or gender	male
date of birth	1971 06 28
occupation	entrepreneur
given name	elon
country of citizenship	united states of america
place of birth	pretoria
educated at	queen 's school of business
award received	honorary degree
family name	musk
instance of	human
relative	lyndon rive
sibling	kimbal musk
languages	english
employer	paypal
discoverer or inventor	hyperloop
spouse	talulah riley
residence	bel air
native language	english
mother	maye musk
member of	the planetary society

Table 7.3: Relations populated for the Elon Musk entity in Wikidata.

7.4 Model

Under the biography generation task, we are able to represent facts naively as a linearized sequence, e.g. "TITLE John Smith GENDER male OCCUPATION painter". This encoding imposes little overhead on the model as we are able to dynamically attend relevant parts of the input sequence via attention. However, a equivalent linearization of facts in the output space has several drawbacks.

First, the order of generated facts should not matter. Moreover, we are unable to accurately estimate model performance for sparsely populated entity relations. When

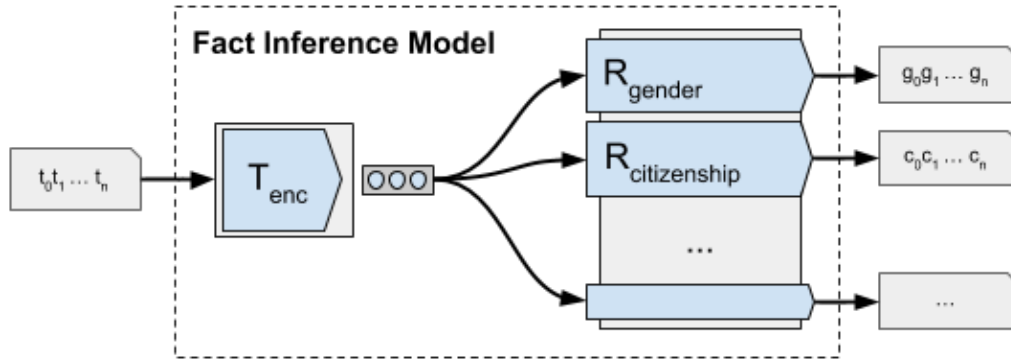


Figure 7.2: A single-input multi-output sequence to sequence fact inference model.

The text encoder network (T_{enc}) produces a shared input sequence representation from which multiple decoders (R_{gender} , $R_{citizenship}$, etc) independently generate target values for each fact type. Vocabularies, sequence length and model parameters are decoupled for each decoder network.

computing error during training over a linearized list of relations, values generated by the model which are missing from the input sequence will be penalized under naive sequence loss, even if they are correct. Finally, linearization of output relations is immensely inefficient. We are unable to take advantage of the significantly reduced output vocabularies for certain fact types (e.g. we need only decode over [male, female] for most gender outputs). Scaling the model to additional fact types also increases the target sequence length and correspondingly the difficulty of backpropagation.

Given these constraints, we opt to introduce multiple fact decoders for each input. Figure 7.2 shows a high level overview of our single-input multi-output translation architecture. We utilize a single shared text encoder T_{enc} which produces an input representation s_m for each input m in the set of mentions for an entity M .

$$s_m = T_{enc}(m)$$

For each target fact we maintain a corresponding decoder model R with independent parameters θ_F . Under the sequence-to-sequence framework each decoder R models the likelihood of an output token f_i as a function of the input mention representation s , model parameters θ_F and previously decoded outputs. We obtain the full sequence

of output tokens for a fact $f_0 f_1 \dots f_n$ via maximum likelihood decoding. The specific structure of this model is described in Section 7.4.1. Here we denote the likelihood of an output sequence for a given input mention and target fact in terms of the tokens of the maximum likelihood decoding:

$$P(f_0 f_1 \dots f_n | s) = \prod_{i=1}^n R(s, [f_0 f_1 \dots f_{n-i}]; \theta_F)$$

For an entity e with multiple input mentions M_e , we decode each fact type F from the mention m_F^* which has highest maximum likelihood sequence probability over all mentions of that entity $m \in M_e$.

$$m_F^* = \underset{m}{\operatorname{argmax}} P(f_0 f_1 \dots f_n | s_m)$$

This architecture has several key advantages over a naive sequence-to-sequence encoding on the fact inference task. At train time, we are able evaluate and apply loss to only those parts of the decoder network for which input facts are populated on each instance. Decoupling of decoder parameters also enables each decoder to attend parts of the input specifically relevant to a specific fact type. At inference time, decoupling the output vocabulary and target sequence length for each fact type greatly reduces the complexity of sequence decoding. In sharing the input encoder network, we are able to take advantage of pooled information under our multi-task decoding objective. This means that a shared input representation is trained jointly over both well and sparsely populated fact targets.

7.4.1 Sequence-to-sequence model

To implement the underlying encoder-decoder model we utilize Transformer networks (Vaswani et al., 2017). Under this model, we first encode sequence inputs via a word embedding, then apply a positional encoding which injects positional information into the representation for each token. Input representations are then propagated through multiple steps of both self-attending (Lin et al., 2017) and fully connected

layers. In contrast to recurrent models considered in Section 6.4, self-attention networks do not explicitly condition model structure on the sequential nature of the input space, allowing for internal states to be computed in parallel and dramatically speeding up both forward and backward propagation steps. This gain in model efficiency facilitates richer modelling of task structure without sacrificing model capacity or run-time — in our case, enabling the introducing of multiple independent decoder networks. For each time-step of the target sequence, we follow a similar method to the base sequence-to-sequence framework. Each decoder takes as input the output embedding for the last generated token, computes an attention state and attends to the output of the encoder network for an instance. At the final layer, we compute softmax over possible tokens in the decoder vocabulary.

7.4.2 Preprocessing

We preprocess each input sequence by first tokenizing the sentence and converting each token to lower case. We utilize fixed, discrete vocabularies for the input encoder and each output fact type — replacing tokens which appear less than twice in the training set with a special out-of-vocabulary identifier oov. Sequence start, end and mention span are also identified by special vocabulary tokens. Under this scheme, the sequence: "A mention of John Smith in text." becomes [BOS, a, mention, of, |, john, smith, |, in, text, ., EOS]. We impose a maximum input sequence length of 35 tokens which covers 80% of mention sequences without truncation. For output sequences, we truncate outputs beyond than 75th percentile of the length distribution for each fact type.

7.4.3 Training

We train our model using the Adam optimizer over mini-batches of 128 entity instances from the training set. To compute a gradient at each mini-batch, we compute the average per-instance loss for each decoder and back-propagate against the aggregate

loss across all decoders. Under this scheme, well-populated fact types (e.g. gender) do not dominate less-populated facts when computing the gradient at each step. We also explore multiple independent gradient steps per decoder each batch but observe overall lower performance in development experiments. After every 1,000 mini-batches we randomly sample and decode 500 instances from the validation set and evaluate micro-averaged fact inference precision across fact types. We select the best performing model under this validation criteria after 50 epochs.

In exploring hyper-parameters configurations, we specify half the number of layers and attention heads for each decoder network as specified for the shared encoder network. As each decoder independently targets a relatively constrained target sequence space in comparison to open domain English, we expect a correspondingly lower model capacity is required in comparison to the encoder network. Moreover, as the number of target facts increases under a given computation budget we are incentivised to push more model complexity into the encoder (which is evaluated once and shared) than decoder networks which must be replicated for each target fact type. We explore a small number of alternative configurations — [128, 256, 512] dimensions for model layer dimensions and [4, 8] for the number of encoder layers. Our final parameterization utilizes 256 dimensions for both word embeddings and hidden layers and 4 layers with 8 attention heads for the encoder and correspondingly 2 layers and 4 attention heads for each decoder.

We explore both Noam learning rate decay as described by Vaswani et al. (2017) and a simple fixed learning rate decay schedule. In development experiments we observe high variance and instability for model configurations under the Noam decay scheme and opt for a static decay factor of 0.99 after a warm-up period of 5K batches. We select the initial learning rate alongside other hyper-parameters within the range $[10^{-3}, 10^{-4}, 10^{-5}]$, with best DEV set performance at 10^{-4} . Under this scheme, model performance generally converges near peak validation set performance after approximately 25k batches.

7.4.4 Inference

Each decoder network is trained to predict an output token given the previously generated tokens and the encoded input sequence representation. We generate output facts one token at a time from left to right until the designated end-of-sequence EOS token is generated. To estimate the maximum likelihood decoding for each fact sequence at test-time we utilize beam search decoding with a beam width of 5. For instances with more than one input mention, we obtain a pool of decoded sequences across each mention from which we select the sequence with the highest decode probability for evaluation.

Under this scheme, we obtain predictions for every fact type at inference time, regardless of whether the input sequence explicitly references the fact or value being queried. In realistic applications of fact inference to KB population, we may wish to suppress output in cases where there is insufficient evidence in the input to resolve a given output fact. To address this issue, we measure how thresholds on the decode probability of inferred facts may be used to trade-off recall for precision in Section 7.5.2.

We may also consider learning to emit a designated `nil` symbol for facts which are not-applicable to an entity — e.g. when predicting the `date_of_death` for a living person. However, under the Wikidata schema we cannot distinguish between instances of unpopulated facts and genuine `nils`. Moreover, we observe that many missing slots under the selected fact schema are cases of missing information, i.e. while we expect all humans to have a valid `place_of_birth`, this slot is only populated for 58% of instances in our dataset (see Table 7.1). As such, we cannot automatically identify true negatives instances within the data and leave exploration of this mechanism to future work.

7.5 Results

We fit two equivalent models to distinct parts of the Wikidata corpus — one taking biographic summary sentences from the entity article as input (BIO), and the other trained over a random sample of up-to 5 sentences with inbound links to the entity page (LNK). As our dataset is split into TRAIN, DEV and TEST at the entity level, both models are evaluated on the same set of held-out entities and differ only in terms of the input sentences they are trained to infer facts from. In evaluating equivalent models across on each dataset we seek to gain some insight into the relative complexity of fact inference from each source.

To asses each model we first consider a detailed precision-oriented evaluation with respect to populated Wikidata facts and the apriori baseline. We then measure how precision and recall vary across thresholds on the likelihood of decoded values.

7.5.1 Comparison with the Wikidata reference

In this section we measure performance of both BIO and LNK models with respect to the reference Wikidata facts. We summarize fact inference precision across fact types for held-out instances from TEST and compare performance across each model.

Results for the BIO model are detailed in Table 7.4 and results for the LNK model are detailed in Table 7.5. We count true positives for instances where the decoded fact exactly matches the Wikidata reference value. For each fact type, we denote **Count** as the number of instances with that fact. **Base** indicates the performance of a baseline system which predicts the most common value for each fact type — e.g. "United States of America" for the `citizenship` slot; see Table 7.1 for a full listing. **System** indicates the performance of the translation model. We list precision of systems over the top ranked output by decode likelihood as **P@1** and over the top-5 outputs as **P@5**. While models are trained and evaluated over the same set of entities, some entities have no inbound

		Base	System	
Fact Type	Count	P	P@1	P@5
sex or gender	139,272	83.5	94.2	99.0
date of birth	118,414	0.2	75.4	80.5
occupation	111,462	11.8	69.8	88.1
given name	110,770	3.4	88.0	94.1
citizenship	102,246	28.1	89.2	94.7
place of birth	81,324	1.5	25.7	36.9
date of death	55,610	0.1	68.3	75.4
place of death	27,618	3.8	27.8	39.2
educated at	25,633	3.7	16.3	33.0
sport	23,067	56.9	87.1	98.1
sports team	21,841	0.5	17.0	31.3
position held	13,953	6.3	63.0	78.8
award received	12,196	4.6	38.8	56.6
family name	11,368	4.4	61.5	70.4
participant of	11,054	6.3	44.5	81.1
political party	10,409	18.3	60.6	83.8
Micro Avg.		20.9	70.0	79.5
Macro Avg.		14.6	58.0	71.3

Table 7.4: Precision of the BIO fact inference model trained and evaluated on biographic summaries for TEST set entities.

links within our Wikipedia extraction — as such, they do not contribute to model evaluation and lead to overall lower counts by fact type in LNK model evaluation.

We observe high precision for fact types inferred from entity summary sentences by the BIO model and performance well above the apriori baseline for all fact types — suggesting our model is able to successfully leverage textual information from the input to infer target facts. In particular, our translation model performs well on fact types such as `sex or gender` which are rarely made explicit in text, requiring the model to infer this information from sources such as the mentioned name. Performance is also high

for facts such as occupation and citizenship which generally necessitate a learned transformation between expressed forms and the Wikidata schema (e.g. "American" into "United States of America").

Interestingly, we observe a positive rank-order correlation ($r = 0.54, p = 0.03$) between instance count and system performance. This may suggest our model benefits from more training instances per fact, or that Wikidata editors are more likely to populate facts for which information is readily available in the entity summary. For facts which are rarely made explicit in the summary sentence and cannot be reliably inferred from other textual information (e.g. `educated at`, `place of death`), we observe overall lower performance.

Comparing performance between the BIO and LNK models, we observe overall lower performance across fact types and per-instance precision reduced to 44.5 from 70.0 for inlink driven fact inference. Encouragingly, LNK model performance for key fact types describing entity identity such as gender, occupation and citizenship remains within 5-20% of results for the BIO model. We expect this gap to be attributable primarily to the information content of input mentions. While biographic summaries utilized as input for the BIO model present a rich source of information, tangential references of entities across Wikipedia mentions are less information dense. For example, we observe poor comparative performance for `date of birth` which is part of the standard format for biographic summaries but can rarely be inferred from inbound mentions. We provide an analysis of fact explicitness for this and other fact types across inbound links in Table 7.8.

By contrast, poor performance on the `family name` fact cannot be explained by a lack of input information as full names are commonly specified in the anchor text string when linking to an entity page. In this case, family names present a particularly difficult case for translation without augmentation via copy-actions or templating as explored in Section 6.4. As embeddings between source and target languages are untied, the model must align symbols in the input language to those replicated in the output even where

		Base	System	
Fact Type	Count	P	P@1	P@5
sex or gender	92,936	83.4	90.2	98.3
date of birth	80,676	0.3	0.3	1.5
occupation	75,154	11.6	50.3	72.4
given name	75,935	3.5	88.5	91.8
citizenship	70,916	31.6	67.7	85.0
place of birth	57,881	1.8	9.1	18.2
date of death	41,407	0.1	0.3	1.0
place of death	22,389	4.0	18.5	30.6
educated at	20,188	3.9	11.7	23.5
sport	12,104	46.6	58.8	80.5
sports team	11,202	0.7	9.3	19.8
position held	10,074	6.7	42.0	60.5
award received	10,051	5.0	20.4	37.9
family name	8,630	4.0	0.0	0.0
participant of	5,857	7.9	20.8	46.0
political party	7,521	18.5	47.6	70.3
Micro Avg.		20.5	44.5	54.6
Macro Avg.		14.4	33.5	46.1

Table 7.5: Precision of the LNK fact inference model trained and evaluated on sentences linking to the entities from the TEST set.

they represent the same token. Family names are both more diverse per-instance and one of the least populated slots within our dataset, yielding fewer instances per-name over which the model can learn a mapping. For the BIO model, family names tend to appear within a fixed region of the input as per the simplified structure of Wikipedia biographies and thus may be easier to recover. By contrast, names may appear at any point within linked mentions. In aggregate, these issues produce a degenerate decoder output of OOV for all instances of this type for the LNK model.

7.5.2 Thresholding decode scores

In this section we explore the precision-recall trade-off for our models on the fact inference task. While each model produces a complete set of outputs for each input sentence, not all entity mentions are good predictors of the target facts. We expect the sequence likelihood of decoded values to be correlated with model confidence, providing a mechanism by which we may threshold and omit decodes for less likely outputs. One potential limitation of this approach is the negative correlation between sequence length and decode probability, i.e. longer outputs are inherently less likely. While this effect may be addressed through the introduction of a length normalization term to beam search scores (Johnson et al., 2017), we expect the significantly shorter target sequence length of facts to mitigate this effect. In practice our results suggest that decoder likelihood provides a useful measure of model confidence.

We sample performance at half-percentile increments across the output likelihood distribution for fact decoders on instances from TEST, obtaining 200 distinct thresholds per fact type. To calculate precision and recall at each threshold, we consider all model outputs below the threshold false negative, all outputs above the threshold which contradict the reference false positive, and all outputs above the threshold and exactly matching the reference true positive. Figure 7.3 details macro-averaged precision and recall across fact types for each model. BIO precision is consistently higher than LNK model results across recall thresholds, with this margin increasing from 0.245 to 0.324 between the lowest and highest decode thresholds. This indicates a better trade-off for precision at low recall levels for fact inference over biographic summaries.

Figure 7.4 plots precision vs recall across fact types for each model. While the rank-order of fact types is broadly consistent across recall thresholds, we note a significant break in this trend for facts such as `award received`. To better understand how our model assigns likelihood at each threshold we extract a sample of mentions for the `award received` fact type in Table 7.6. Encouraging, we observe mentions from the

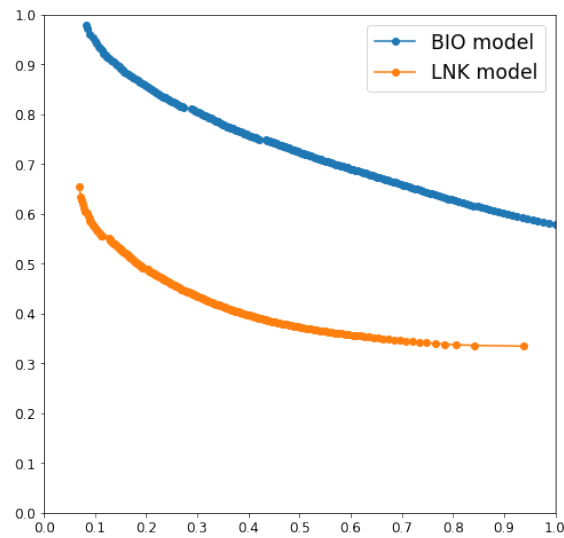


Figure 7.3: Macro-averaged Precision vs Recall for each model.

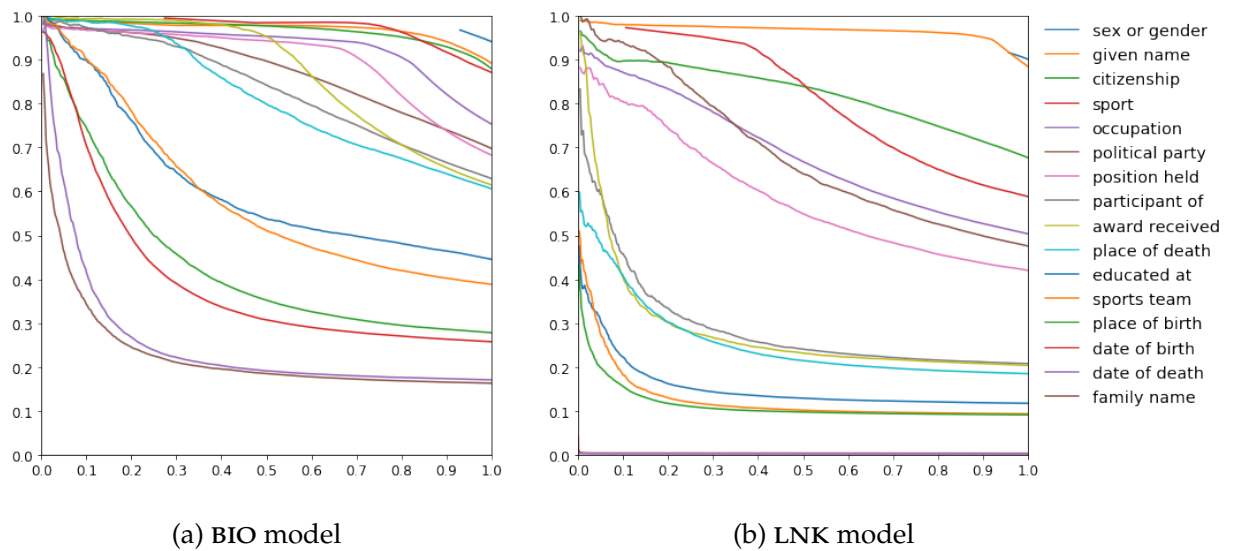


Figure 7.4: Precision vs Recall across fact-types

Pctl.	Mention	Reference	System
50th	george robert stibitz (april 30 , 1904 - january 31 , 1995) is internationally recognized as one of the fathers of the modern first digital computer.	national inventors hall of fame	ieee medal of honor
75th	vera nikolaevna maslennikova (; 29 april 1926 - 14 august 2000) was a russian mathematician known for her contributions to the theory of partial differential equations.	order of the patriotic war 2nd class	ussr state prize
95th	karl pearson frs (; originally named carl ; 27 march 1857 - 27 april 1936) was an influential english mathematician and biostatistician.	fellow of the royal society	fellow of the royal society
99th	yvonne mcgregor mbe (born 9 april 1961) is an english former professional cyclist from wibsey.	member of the order of the british empire	member of the order of the british empire

Table 7.6: Sampled mentions across confidence thresholds for the award received fact type. **Pctl.** indicates the position of each mention within the decode likelihood distribution; higher percentiles being more likely. **Reference** indicates the Wikidata reference value and **System** indicates the output of the BIO model.

lower percentiles of the likelihood distribution provide poor evidence for the predicted fact value, while more reasonable assignments obtain higher likelihoods (e.g. a famous English mathematician is likely to be a fellow of the royal society). At the 99th percentile, we observe direct evidence for the predicted fact value (i.e. the presence of the MBE honorific). Broadly, a sharper knee in the curve for relational fact types may indicate facts for which more direct evidence is required to predict a value (e.g. place of birth, sports team and award received). We carry out a deeper analysis of fact expression over LNK model mentions in Section 7.6.3.

7.6 Analysis

7.6.1 Performance vs Inlink Count

In this section we consider the relationship between the precision of generated facts and the number of input sentences available. While our trained model may be applied to any number of input mentions without modification, our analysis in this section is limited by our construction of the dataset to samples of up-to 5 mentions per entity. Still we hope to give some indication of how performance may scale to larger samples.

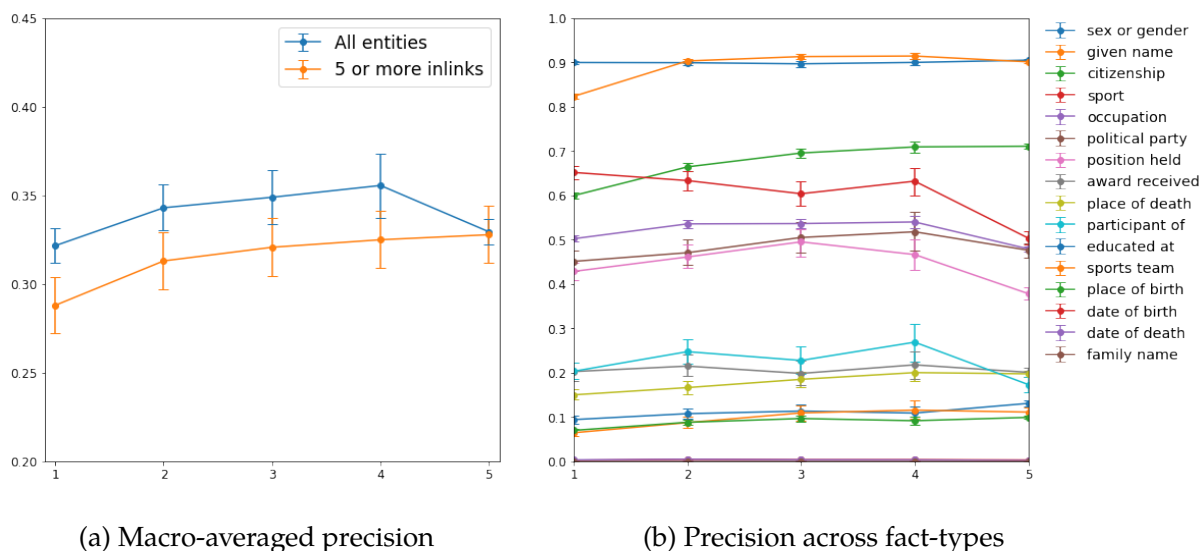


Figure 7.5: Figure (a) indicates macro-averaged precision vs inlink count on alternative instance subsets. Figure (b) indicates precision vs inlink count across fact-types for all TEST entities. Metrics are calculated over LNK model output on held-out TEST set entities. Error bars indicate the bootstrapped 95% confidence interval over 1,000 samples.

We detail performance vs inlink count in Figure 7.5. While we observe a moderate but statistically significant increase in performance from 1 to 4 mentions across fact types, we note a significant decrease in precision for instances with five source inlinks. As we describe in Section 7.3, each instance contains a random sample of **up-to** five

mentions per entity. As such, instances with 5 mentions represent entities with five or more distinct mentions across Wikipedia, a relatively notable and potentially distinct population relative to those with 4 or less inbound links. To better control for this effect, we separately measure performance vs inlink count over these instances in Figure 7.5a by randomly sub-sampling inlinks across these instances. Here we observe that while precision is generally lower across these entities, overall performance increases with mention count as expected. This suggests that fact inference is moderately easier for less notable entities within Wikipedia. Intuitively, we speculate that less notable entities may be described in more detail via linked mentions as authors assume a reader may not otherwise be familiar with the referenced entity. While interesting, this effect is small and deeper analysis is required to account for this disparity.

7.6.2 Example generated facts

To better understand how facts are decoded across input sentences for each entity, we provide a decoding of the 5 most common fact types across distinct mentions for a set of sample entities in Table 7.7. Here we highlight interesting aspects of LNK model performance. In the case of Harold Theobald, our model correctly infers all facts except date of birth (DOB), despite only having a single input mention from which to extract information. While only the entity name Harold is made explicit in text, inferring an occupation of cricketer is certainly justified given the observed reference to participation in a "wicket partnership". Moreover, given the popularity of cricket within the United Kingdom, predicting this state for citizenship is a reasonable and in this case correct guess — albeit unjustified by the text.

In the case of Dan Hardy, we may consider how alternative decodes are generated for the same fact across distinct mentions. In particular, the model predicts different occupations including "mixed martial artist", "association football player" and "boxer" across mentions. These alternatives may be explained in turn by textual references to terms associated with by not exclusive to those occupations, i.e. respectively: "UFC",

"coach" and "fight" / "welterweight". Interestingly none of these decodes match the Wikidata reference for this entity "thai boxer", despite close association with this term. This suggests some accounting of distance in terms of hypernymy and fact specificity during evaluation may give a better estimate of model performance.

As we will show in Section 7.6.3, it is rarely possible to infer an exact date of birth from isolated mentions of an entity in Wikipedia. However, guesses made by the model within a range of the reference date often appear reasonable. We observe a strong positive correlation ($r = 0.78, p < 0.001$) between the decoded birth year and corresponding Wikidata reference across instances from TEST with a well-formed decode date. While `date of birth` is among the worst performing facts types under an exact-match criterion (see Table 7.5), this analysis shows a median difference of just 12 years between the predicted year and reference birth date for these instances. This correlation suggests our model can make use of basic associations between expressed information and entity age in cases where age itself is not expressed. For example, in the case of Vladimir Ryzhenkov we may infer from the source sentence that the target is a government minister participating in events around 1993 and therefore likely to be middle aged at the time. Given this information, predicting a `date of birth` from the 1950's appears reasonable. We expect simple textual associations between expressed information and the target birth year to account for an otherwise impressive ability to emit well correlated birth dates.

	name	occupation	citizenship	gender	DOB
Harold Theobald	harold	cricketer	united kingdom	male	1896 03 18
in a minor counties record first-wicket partnership of 323 with harold theobald .	harold	cricketer	united kingdom	male	1912 01 01
Dan Hardy	dan	thai boxer	united kingdom	male	1982 05 17
ludwig fought dan hardy on may 26 , 2012 , at ufc 146 .	dan	mixed martial artist	united states of america	male	1985 01 01
first mma events , the ksbo , where they met dan hardy and coach nathan leverton .	dan	association football player	united states of america	male	1976 01 01
he was scheduled to fight dan hardy on november 14 , 2009 at ufc 105 , but	dan	mixed martial artist	united states of america	male	1983 01 01
fights before going on a three fight losing streak against dan hardy , brandon wolff and tj grant .	dan	mixed martial artist	united states of america	male	1983 01 01
in his post fight interview , he called out dan hardy , the # 1 contender in the welterweight division .	dan	boxer	united states of america	male	1978 01 01
Cecilia Sigurdsdottir	cecilia	nil	norway	female	1200 01 01
folkvid the lawspeaker and king sigurd munn 's illegitimate daughter cecilia sigurdsdottir	cecilia	politician	norway	female	1890 01 01
he is only mentioned by virtue of his marriage to cecilia , the bastard daughter of the norwegian king sigurd munn	cecilia	politician	norway	female	1200 01 01
Vladimir Ryzhenkov	wladimir	weightlifter	russia	male	1948 08 27
also in 1993 , vladimir ryzhenkov , who was at the time the belarus minister for	vladimir	politician	russia	male	1952 01 01

Table 7.7: Sampled mentions and corresponding facts decoded for selected entities. Each section denotes an entity where the header row indicates the entity title in **bold** and corresponding gold-standard Wikidata facts for that entity across adjacent columns. The following rows contain sampled mentions and the corresponding facts decoded by the model. We denote the maximum likelihood decoding for an entity-fact pair in standard font, and use grey for lower scoring decodes.

7.6.3 Fact Explicitness

In this section we attempt to quantify the disparity between the information content of textual entity mentions and target facts. While we have previously suggested this mismatch exists and observed specific cases within decoded samples, we seek a more rigorous evaluation of fact explicitness. We randomly sample 50 source sentence-fact pairs for each of the top-5 populated fact types and annotate the degree to which a human annotator can infer the reference fact value from the provided source text.

A discrete categorization of fact explicitness is challenging. Pink (2017) analyze the explicitness of expressed relations within the closely related task of Slot Filling, finding that 20% of annotated relations may be removed from a standard dataset under a strict definition of explicitness. They note a large scope for disagreement between annotators exists based on differences in prior world knowledge and variation in decision thresholds for probabilistic inference. While we utilize their explicitness annotation scheme as a guide, a direct mapping from SF is inappropriate under our formulation of the fact inference task.

We categorize fact expression according to the following scheme:

- **Explicit** - where the value of a fact is explicitly realized in text, e.g. mentioning "Canadian" or "born in Canada" for a citizenship value of "Canada".
- **Reasonable** - where the value of a fact is directly implied by statements in text, e.g. mentioning the referent is "born in London" for a citizenship value of "United Kingdom".
- **Guessable** - where the reference value for a fact may be a reasonable guess given the source text, e.g. the value of "film actor" when the entities is said to "star as the main protagonist" in a series, though other values are justifiable (e.g. "television actor").
- **Unjustified** - where the source text provides no evidence for a specific fact value.

Fact Type	Explicit	Reasonable	Guessable	Unjustified
given name	93.9	0.0	0.0	6.1
occupation	2.2	46.7	35.6	15.6
citizenship	6.7	6.7	20.0	66.7
sex or gender	4.4	84.4	4.4	6.7
date of birth	6.5	0.0	0.0	93.5
All Types	23.9	27.0	11.7	37.4

Table 7.8: Analysis to fact expression across inlinks to entity pages.

We summarize results across annotated instances in Table 7.8. All instances are judged by a single annotator, i.e. the author. In many cases it is difficult to robustly distinguish between cases of "Reasonable" or "Guessable" facts. We observe that LNK model performance is roughly aligned with the proportion of Explicit and Reasonable annotations across fact types excluding citizenship. In the case of citizenship, the model outperforms this baseline — possibly by taking advantage of correlations between last names and citizenship or the high prior for "United States of America".

7.7 Discussion

Translation models are a powerful mechanism for transforming information between alternative representations. On the fact inference task, we demonstrate the capability for these models to both learn the target KB schema and infer the value of facts which may never be explicitly realized in text. Even when this information is generally missing even in latent form (e.g. the year an entity was born), the ability to automatically extract and learn associations between relevant information from the source text and emit a well-correlated guess is compelling.

While we observe generally strong performance across fact types for biographic summaries, certain fact types remain problematic for a vanilla translation model. In particular, we expect that augmenting our sequence-to-sequence framework with copy

actions (Vinyals et al., 2015a; Gu et al., 2016; Jia and Liang, 2016) will significantly reduce the complexity of decoding for large-vocab fact types such as `family name`, `educated at` and `sports team`. In these cases, the model can learn to utilize input text verbatim when it appears both in the input and target fact value — a mechanism bridging the gap between fact inference and extractive task formulations. We also expect initialisation of source and target vocabularies from pretrained word embeddings (e.g. GloVe; Pennington et al. (2014)) to improve performance, especially for sparsely represented tokens or output fact variations over which our model has fewer samples to reliably infer a relation. Alternatively, we may mitigate vocabulary constraints by adopting a common sub-word representation between encoder and decoder, e.g. via byte pair encoding techniques which implicitly capture word morphology and structure (Sennrich et al., 2015).

For experiments involving multiple input mentions, we adopt a maximum likelihood decoding scheme which pools isolated decodes across distinct mentions. Even under this simple scheme we demonstrate an increase in fact inference precision as the number of sampled input mentions grows. Still, better aggregation of information across mentions may further improve performance. We may pool information at the system level via likelihood-weighted voting over decoded values or at the model level through the integration of a hierarchical attention mechanism (Yang et al., 2016) over encoded inputs. While the former approach precludes the possibility of reasoning across input sentences, it maintains the horizontal scalability of our fact inference model across multiple entity mentions — an important characteristic for models targeting web-scale corpora.

Our experiments consider a small set of commonly expressed facts types. Scaling up the multi-decoder framework to a broader set of output fact types presents a variety of potential challenges. While we may address increased computational complexity by evaluating decoders independently and in parallel, model size still grows linearly with the number of fact types being decoded. Moreover, as we observe fewer training

instances for sparsely populated KB facts it may become increasingly difficult to fit a robust decoder model for each fact type. We expect models which condition decoding on the target fact type without introducing additional parameters (e.g. Hewlett et al. (2016) feed the target fact as an input to a shared decoder) and better take advantage of similarities in the way different relations are expressed are a key direction for future work.

Breaking the tight coupling between the surface form of expressed facts and KB values offers some advantages, but also breaks the direct link between evidence for a fact in text and knowledge persisted to the KB. Our model can identify the sentence which produces the most likely decoding for an output fact¹ but cannot describe the reasoning behind a given inference. Applications of inference to domains where justified predictions are crucial will necessitate deeper model introspection (Ribeiro et al., 2016). In addition, our analysis of fact expression suggests that the model leans on priors from the KB, raising potential concerns around the perpetuation of machine learned biases (Barocas and Selbst, 2016) from the unrepresentative Wikidata population (e.g. 84% of entities are male, and worse still, nearly one in eight are politicians) (Wagner et al., 2015).

7.8 Summary

In this chapter we adapt the knowledge translation framework to the fact inference task. Our multi-output sequence transformer model is able to infer key facts about entity identity, even where those facts are not explicitly described in text. In comparison with existing approaches to inference over Wikipedia articles (Hewlett et al., 2016), we extend our framework to inbound links for an entity page and contribute a detailed analysis of new models on this data.

¹Sufficient for Wikidata review: https://www.wikidata.org/wiki/Wikidata:Primary_sources_tool

While our exploration and framing of this task is motivated by applications to knowledge extracted from web KBs, our experiments are constrained to evaluation on Wikipedia and Wikidata resources — leaving applications to the web IE setting as a clear path for future work. Putting this direction aside, we still expect our models to be useful in the traditional KB setting, where both text generation and fact inference have great potential for use in KB population and curation.

8 Conclusion

Knowledge of and about entities is central to natural language understanding. Dedicated stores of entity knowledge enable applications in downstream tasks like entity based search and question answering, but are expensive to scale, maintain and update over time. The web presents an alternative source of entity information, but necessitates a mechanism for extracting useful entity knowledge from an otherwise inscrutable assortment of unstructured data.

This thesis explores the idea that many sites on the web may *act* as a KB, even if that is not their primary intent. Where pages represent or describe specific real world entities, they often incidentally serve as disambiguation endpoints for inbound links across the web. This KB-like structure provides a direct mechanism for extracting disambiguated entity mentions from text on the web which in turn simplify downstream information extraction tasks. In Chapter 3 we utilize inlinks to Wikipedia as a source of knowledge in named entity disambiguation. Our first core contribution is the development of a NED system incorporating KB and web link derived disambiguation features. Our experiments show that inlinks to a KB can both complement and in some cases completely replace resources of the KB itself in NED, with a combination of both improving the state of the art. In the chapters which follow we build upon this insight, attempting to generalize the use of web linked entity resources to KB-like endpoints beyond Wikipedia.

Chapter 4 investigates the task of Knowledge Base Discovery (KBD) — finding endpoints on the web which disambiguate inbound web links. Our core contribution

is to formalize this task and develop a method for classifying candidate endpoint links using mention-URL cooccurrence over a corpus of web linked documents. We develop an annotated dataset for KBD evaluation and analyze the results of KBD for a corpus of web news documents — demonstrating that a wide variety of resources index entities on the web, many of which are not otherwise covered in dedicated KBs like Wikipedia. In uncovering these resources, we observe that many distinct web entity endpoints reference the same underlying entities. To consolidate coreferent entity URLs across web KBs we explore Cross-KB Coreference Resolution (KB-Coref) in Chapter 5. We extend our KBD experiments to two new large-scale web documents collections and additionally merge discovered entity URLs into clusters of coreferent links via a distributed hierarchical agglomerative clustering system.

Entity endpoints facilitate the aggregation of unstructured entity knowledge through textual mentions on the web. These resources provide both broader and deeper entity coverage than a standalone dedicated-KB, but lack analogues for structured factual knowledge and entity summaries which make otherwise unstructured information useful to downstream systems and human consumers alike. In the chapter 6, we address this issue in part by developing a framework for knowledge translation. Our primary contribution is a system taking Wikidata facts as input and producing single sentence Wikipedia-style biographic descriptions as output. We provide an analysis of fact-driven text generation — analyzing human preference and fact-level content selection alongside standard translation metrics. Our system is able to produce entity biographies nearly indistinguishable to a Wikipedia reference in terms of fluency, but cannot always precisely convey the supplied input facts. In the final chapter, we target the inverse task of inferring facts from mentions of an entity in text. Following the translation framework developed for biography generation, we show that a fact inference model is able to both infer the value of implicitly described facts from text and learn to canonicalize fact values under a target KB schema. Our evaluation over Wikidata facts shows that our model can recover facts such as gender, citizenship and occupation

with high precision from both biographic entity descriptions and inbound links to an entity across Wikipedia.

8.1 Future Work

Each chapter in this thesis describes specific extensions to the task and methods discussed therein. In this section we consider avenues for future work combining and extending components of this framework as a whole. In Chapter 3 we develop a named entity disambiguation system with Wikipedia as a target for entity resolution. Given the demonstrated feasibility of link-driven disambiguation modelling, work described in subsequent chapters on discovering and clustering alternative web entity endpoints presents a clear path for extending NED. In place of a single dedicated KB, we may instead resolve document mentions to clusters of coreferent entity URLs — benefiting from both wider entity coverage and richer disambiguation context.

Within document entity resolution has further applications throughout the web knowledge extraction pipeline. While we generally only observe a single linked entity mention for a given web document, there are often many subsequent coreferential named or pronominal references to an entity throughout document text. Our experiments only consider link-annotated entity mentions in entity modelling, suggesting that the incorporation of in-document coreference and NED may greatly increase the number of mentions extracted beyond those those explicitly annotated with outbound web links. We expect the incorporation of these resources will further enrich downstream knowledge extraction tasks.

While deep learning models utilized in Chapters 6 and 7 provide a powerful mechanism for modelling unstructured natural language, we do not apply these methods to other applicable tasks considered in this thesis. In particular, we expect the mention-modelling tasks considered in Chapters 3 and 5 may benefit significantly from learned representations which embed disambiguating entity information, as has been demon-

strated in recent work (Huang et al., 2015; Clark and Manning, 2016). We account for this discrepancy in part by noting the scale at which web-based IE systems must operate. In Chapters 4 and 5 we utilize efficient linear models which impose a low computational overhead and easily scale to corpora with billions of documents and links at inference time. We expect ongoing progress in computation and model efficiency to benefit the framework described in this thesis through the application of higher-capacity models alone — in particular, the techniques we develop for heuristically sampling training instances in KBD and KB-Coref experiments enable extraction of large-scale supervised machine learning datasets without a correspondingly large cost in data annotation. Combining these techniques with high-capacity machine learning models able to take advantage of these resources is a clear avenue for future work.

In the two chapters we consider transformations of entity knowledge between structured and unstructured forms. We specifically address fact-to-text biography generation and text-to-fact inference, though many other input-output configurations are possible. Johnson et al. (2017) demonstrate that joint training of language translation models across multiple language pairs improves collective translation performance, even providing a mechanism for performing zero-shot translation across pairs not observed during training. We expect that joint training across fact and text generation tasks, in addition to the autoencoding objective we have already explored will further improve translation performance. Moreover, the incorporation of additional translation modalities as explored by Kaiser et al. (2017) presents an interesting avenue for future work. For example, we may learn to predict facts about an entity given an image, or even generate imagery (Reed et al., 2016; Mansimov et al., 2016) given knowledge extracted from facts and textual mentions.

8.2 Summary

In total, this thesis explores the key components in finding, extracting and developing web based sources of entity knowledge. We introduce an inlink driven system for entity disambiguation with Wikipedia, then generalize this approach to model entities in terms of inlinks to KB-like structures across the web as a whole. We develop a system for clustering coreferent entity endpoints across web KBs and develop the pipeline for extracting entity information from large-scale open-access web document collections over time. We then address the gap between structured and unstructured knowledge resources by building models which translate equivalent representations of entity information between the two.

Web sourced entity information has great potential as a source of knowledge in artificial intelligence tasks. While much work remains in developing the potential of these resources in general, inlinks to KB-like structures present a direct opportunity for aggregating entity information. This thesis lays the groundwork for extracting useful knowledge from links to entity pages across text on the web.

Bibliography

- Heike Adel, Benjamin Roth, and Hinrich Schütze. 2016. Comparing convolutional neural networks to traditional models for slot filling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 828–838.
- Shubham Agarwal and Marc Dymetman. 2017. A surprisingly effective out-of-the-box char2char model on the E2E NLG challenge dataset. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 158–163.
- Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pages 85–94.
- Ayman Alhelbawy and Robert Gaizauskas. 2014. Graph ranking for collective named entity disambiguation. In *Annual Meeting of the Association for Computational Linguistics*, pages 75–80.
- Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Conference on Empirical Methods in Natural Language Processing*, pages 502–512.
- Gabor Angeli and Christopher Manning. 2013. Philosophers are mortal: Inferring the truth of unseen facts. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 133–142.

- Chinatsu Aone, Lauren Halverson, Tom Hampton, and Mila Ramos-Santacruz. 1998. SRA: Description of the IE2 system used for MUC-7. In *Proceedings of the 7th Message Understanding Conference*, pages 123–135.
- Javier Artiles, Andrew Borthwick, Julio Gonzalo, Satoshi Sekine, and Enrique Amigó. 2010. WePS-3 Evaluation Campaign: Overview of the Web People Search Clustering and Attribute Extraction Tasks. In *Conference and Labs of the Evaluation Forum 2010 LABs and Workshops, Notebook Papers*.
- 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 4th International Workshop on Semantic Evaluations*, 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 2nd Web People Search Evaluation Workshop, 18th WWW Conference*.
- Javier Artiles, Julio Gonzalo, and Felisa Verdejo. 2005. A testbed for people searching strategies in the WWW. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 569–570.
- Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web Conference*, pages 722–735.
- Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 1*, pages 79–85.
- David Bamman and Noah A. Smith. 2014. Unsupervised discovery of biographical structure from text. *Transactions of the Association for Computational Linguistics*, 2:363–376.

- Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2670–2676.
- Solon Barocas and Andrew D. Selbst. 2016. Big Data’s Disparate Impact. *California Law Review*, 104:671.
- Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don’t count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Annual Meeting of the Association for Computational Linguistics*, pages 238–247.
- Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.
- Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2007. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems 19*, pages 153–160.
- Michael K. Bergman. 2001. White paper: The deep web. surfacing hidden value. *The Journal of Electronic Publishing*, 7(1):online.
- Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The semantic web. *Scientific American*, 284(5):34–43.
- Fadi Biadisy, Julia Hirschberg, and Elena Filatova. 2008. An unsupervised approach to biography production using Wikipedia. In *Annual Meeting of the Association for Computational Linguistics*, pages 807–815.
- Lidong Bing, Mingyang Ling, Richard C. Wang, and William W. Cohen. 2016. Distant IE by bootstrapping using lists and document structure. In *AAAI*. To appear.

- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media.
- Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. 2008. Linked data on the web (LDOW2008). In *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, pages 1265–1266.
- Sasha Blair-Goldensohn, David Evans, Vasileios Hatzivassiloglou, Kathleen McKeown, Ani Nenkova, Rebecca Passonneau, Barry Schiffman, Andrew Schlaikjer, Advait Siddharthan, and Sergey Siegelman. 2004. Columbia University at DUC 2004. In *Proceedings of the Document Understanding Workshop*, pages 23–30.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1250.
- Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 301–306.
- Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 10–21.
- Sergey Brin. 1999. Extracting patterns and relations from the world wide web. In *Selected Papers from the International Workshop on The World Wide Web and Databases*, pages 172–183.

- Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. 2000. Graph structure in the web. In *Proceedings of the 9th International World Wide Web Conference on Computer Networks : The International Journal of Computer and Telecommunications Networking*, pages 309–320.
- Razvan Bunescu and Marius Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 9–16.
- Anaïs Cadilhac, Andrew Chisholm, Ben Hachey, and Sadegh Kharazmi. 2015. Hugo: Entity-based news search and summarisation. In *CIKM Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 51–54.
- Michael J. Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. Webtables: Exploring the power of tables on the web. *Proceedings of the VLDB Endowment*, 1(1):538–549.
- Michael J. Cafarella, Jayant Madhavan, and Alon Halevy. 2009. Web-scale extraction of structured data. *SIGMOD Rec.*, 37(4):55–61.
- Rui Cai, Xiaodong Zhang, and Houfeng Wang. 2016. Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.
- Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, Jr., and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 1306–1313.

- Yin-Wen Chang, Cho-Jui Hsieh, Kai-Wei Chang, Michael Ringgaard, and Chih-Jen Lin. 2010. Training and testing low-degree polynomial data mappings via linear SVM. *Journal of Machine Learning Research*, 11:1471–1490.
- Gong Cheng, Danyun Xu, and Yuzhong Qu. 2015. Summarizing entity descriptions for effective and efficient human-centered entity linking. In *International Conference on World Wide Web*, pages 184–194.
- Andrew Chisholm and Ben Hachey. 2015. Entity disambiguation with web links. *Transactions of the Association for Computational Linguistics*, 3:145–156.
- Andrew Chisholm, Ben Hachey, and Diego Mollá. 2016a. Overview of the 2016 ALTA shared task: Cross-KB coreference. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 161–164.
- Andrew Chisholm, Will Radford, and Ben Hachey. 2016b. Discovering entity knowledge bases on the web. In *NAACL Workshop on Automated Knowledge Base Construction*, pages 7–11.
- Andrew Chisholm, Will Radford, and Ben Hachey. 2017. Learning to generate one-sentence biographies from wikidata. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 633–642.
- Kevin Clark and Christopher D. Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.
- Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167.

- Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam, and Seán Slattery. 1998. Learning to extract symbolic knowledge from the world wide web. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, pages 509–516.
- Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 708–716.
- Silviu Cucerzan. 2011. TAC entity linking by performing full-document entity extraction and disambiguation. In *Proceedings of the 2011 Text Analysis Conference*.
- James R. Curran, Tara Murphy, and Bernhard Scholz. 2007. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 172–180.
- Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In *Annual Conference on Neural Information Processing Systems*, pages 3079–3087.
- Jeffrey Dalton and Laura Dietz. 2013. UMass CIIR at TAC KBP 2013 entity linking: query expansion using Urban Dictionary. In *Proceedings of the 2013 Text Analysis Conference*.
- Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43.
- Xin Dong, Evgeniy Gabrilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmman, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 601–610.

Pablo Ariel Duboue and Kathleen R McKeown. 2003. Statistical acquisition of content selection rules for natural language generation. In *Conference on Empirical Methods in Natural Language Processing*, pages 121–128.

Daniel Duma and Ewan Klein. 2013. Generating natural language from linked data: Unsupervised template extraction. In *International Conference on Computational Semantics*, pages 83–94.

Long Duong, Hadi Afshar, Dominique Estival, Glen Pink, Philip Cohen, and Mark Johnson. 2018. Active learning for deep semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 43–48.

Jeffrey L. Elman. 1990. Finding structure in time. *COGNITIVE SCIENCE*, 14(2):179–211.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.

Jérôme Euzenat and Pavel Shvaiko. 2007. *Ontology Matching*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545.

Ivan P Fellegi and Alan B Sunter. 1969. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210.

Paolo Ferragina and Ugo Scaiella. 2010. TAGME: On-the-fly annotation of short text fragments (by Wikipedia entities). In *International Conference on Information and Knowledge Management*, pages 1625–1628.

- Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. FACC1: Freebase annotation of ClueWeb corpora, Version 1.
- William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. One sense per discourse. In *Proceedings of the Workshop on Speech and Natural Language*, pages 233–237.
- Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619 – 2629.
- Alberto García-Durán, Antoine Bordes, Nicolas Usunier, and Yves Grandvalet. 2016. Combining two and three-way embedding models for link prediction in knowledge bases. *Journal of Artificial Intelligence Research*, 55(1):715–742.
- Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Nikesh Garera and David Yarowsky. 2009. Structural, transitive and latent models for biographic fact extraction. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 300–308.
- Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. Multilingual language processing from bytes. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1296–1306.
- Christoph Goller and Andreas Küchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of the 1996 International Conference on Neural Networks*, pages 347–352.
- Chung Heong Gooi and James Allan. 2004. Cross-document coreference on a large scale corpus. In *HLT-NAACL 2004: Main Proceedings*, pages 9–16.

- Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. 2013. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, pages 6645–6649.
- Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference-6: A Brief History. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, pages 466–471.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.
- Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named entity recognition in query. In *International Conference on Research and Development in Information Retrieval*, pages 267–274.
- Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To link or not to link? a study on end-to-end tweet entity linking. In *HLT-NAACL*, pages 1020–1030.
- Bikash Gyawali and Claire Gardent. 2014. Surface realisation from knowledge-bases. In *Annual Meeting of the Association for Computational Linguistics*, pages 424–434.
- Christian Hachenberg and Thomas Gottron. 2012. Finding good URLs: Aligning entities in knowledge bases with public web document representations. In *ISWC Workshop on Linked Entities*, pages 17–28.
- Ben Hachey, Joel Nothman, and Will Radford. 2014. Cheap and easy entity evaluation. In *Annual Meeting of the Association for Computational Linguistics*, pages 464–469.
- 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: Volume 3*, 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.
- Sangdo Han, Jeesoo Bang, Seonghan Ryu, and Gary Geunbae Lee. 2015. Exploiting knowledge base to generate responses for natural language dialog listening agents. In *Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 129–133.
- Xianpei Han and Le Sun. 2011. A generative entity-mention model for linking entities with knowledge base. In *Annual Meeting of the Association for Computational Linguistics*, pages 945–954.
- Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th International Conference on Research and Development in Information Retrieval*, pages 765–774.
- 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.
- Zhiyuan Liu Maosong Sun Hao Zhu, Ruobing Xie. 2017. Iterative entity alignment via joint knowledge embeddings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4258–4264.
- Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.
- 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, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013b. Learning entity representation for entity disambiguation. In *Annual Meeting of the Association for Computational Linguistics*, pages 30–34.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Workshop on Statistical Machine Translation*, pages 187–197.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING*, pages 539–545.

Daniel Hewlett, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot. 2016. WIKIREADING: A novel large-scale language understanding task over Wikipedia. In *Proceedings of the The 54th Annual Meeting of the Association for Computational Linguistics*.

Geoffrey E Hinton. 1986. Learning distributed representations of concepts. In *Proceedings of the 8th Annual Conference of the Cognitive Science Society*, pages 1–12.

Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Johannes Hoffart, Yasemin Altun, and Gerhard Weikum. 2014. Discovering emerging entities with ambiguous names. In *International World Wide Web Conference*, pages 385–396.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 194:28–61.

- 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 *Conference on Empirical Methods in Natural Language Processing*, pages 782–792.
- Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882.
- Hongzhao Huang, Larry Heck, and Heng Ji. 2015. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *CoRR*, abs/1504.07678.
- Lifu Huang, Avirup Sil, Heng Ji, and Radu Florian. 2017. Improving slot filling performance with attentive neural networks on dependency structures. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2578–2587.
- Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Annual Meeting of the Association for Computational Linguistics*, pages 1148–1158.
- Heng Ji, Ralph Grishman, and Hoa Trang Dang. 2011. Overview of the TAC 2011 knowledge base population track. In *Proceedings of the 2011 Text Analysis Conference*.
- Heng Ji, Joel Nothman, Ben Hachey, and Radu Florian. 2015. Overview of TAC-KBP2015 tri-lingual entity discovery and linking. In *Proceedings of the 2015 Text Analysis Conference*.
- Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22.

- Yuzhe Jin, Emre Kıcıman, Kuansan Wang, and Ricky Loynd. 2014. Entity linking at the tail: Sparse signals, unknown entities, and phrase models. In *International Conference on Web Search and Data Mining*, pages 453–462.
- Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *International Conference on Knowledge Discovery and Data Mining*, pages 133–142.
- Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association of Computational Linguistics*, 5:339–351.
- Lukasz Kaiser, Aidan N. Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. 2017. One model to learn them all. *CoRR*, abs/1706.05137.
- Jun’ichi Kazama and Kentaro Torisawa. 2007. Exploiting wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707.
- Gitansh Khirbat, Jianzhong Qi, and Rui Zhang. 2016. Disambiguating entities referred by web endpoints using tree ensembles. In *Proceedings of the 7th Australasian Language Technology Association Workshop*.
- Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. *CoRR*, abs/1312.6114.
- Stanley Kok and Pedro Domingos. 2007. Statistical predicate invention. In *Proceedings of the 24th International Conference on Machine Learning*, pages 433–440.

- Ravi Kondadadi, Blake Howald, and Frank Schilder. 2013. A statistical NLG framework for aggregated planning and realization. In *Annual Meeting of the Association for Computational Linguistics*, pages 1406–1415.
- Ioannis Konstas and Mirella Lapata. 2012. Concept-to-text generation via discriminative reranking. In *Annual Meeting of the Association for Computational Linguistics*, pages 369–378.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105.
- Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1378–1387.
- Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning - Volume 37*, pages 957–966.
- Ni Lao, Amarnag Subramanya, Fernando Pereira, and William W. Cohen. 2012. Reading the web with learned syntactic-semantic inference rules. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1017–1026.
- Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213.
- Yann LeCun and Yoshua Bengio. 1998. *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge, MA, USA.

- Douglas B. Lenat. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.
- Yang Li, Chi Wang, Fangqiu Han, Jiawei Han, Dan Roth, and Xifeng Yan. 2013. Mining evidences for named entity disambiguation. In *International Conference on Knowledge Discovery and Data Mining*, pages 1070–1078.
- Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 71–78.
- Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2181–2187.
- Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.
- Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *Proceedings of the 2017 International Conference on Learning Representations*.
- Xiao Ling, Sameer Singh, and Daniel S. Weld. 2015. Design challenges for entity linking. *Transactions of the Association of Computational Linguistics*, 3:315–328.
- H. Liu and P. Singh. 2004. ConceptNet — A practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226.
- Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *Proceedings of the 2016 International Conference on Learning Representations*.

- Jayant Madhavan, David Ko, Lucja Kot, Vignesh Ganapathy, Alex Rasmussen, and Alon Halevy. 2008. Google's deep web crawl. *Proceedings of the VLDB Endowment*, 1(2):1241–1252.
- Gideon S. Mann and David Yarowsky. 2003. Unsupervised personal name disambiguation. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, pages 33–40.
- Elman Mansimov, Emilio Parisotto, Jimmy Ba, and Ruslan Salakhutdinov. 2016. Generating images from captions with attention. In *Proceedings of the 2016 International Conference on Learning Representations*.
- Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534.
- Paul McNamee, Heather Simpson, and Hoa Trang Dang. 2009. Overview of the TAC 2009 Knowledge Base Population Track. In *Proceedings of the 2009 Text Analysis Conference*.
- Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2015. What to talk about and how? Selective generation using LSTMs with coarse-to-fine alignment. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 720–730.
- Edgar Meij, Wouter Weerkamp, and Maarten de Rijke. 2012a. Adding semantics to microblog posts. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, pages 563–572.
- Edgar Meij, Wouter Weerkamp, and Maarten de Rijke. 2012b. Adding semantics to microblog posts. In *International Conference on Web Search and Data Mining*, pages 563–572.

- Pablo N. Mendes, Max Jakob, and Christian Bizer. 2012. DBpedia: A multilingual cross-domain knowledge base. In *International Conference on Language Resources and Evaluation*, pages 1813–1817.
- Roberta Merchant, Mary Ellen Okurowski, and Nancy Chinchor. 1996. The multilingual entity task (MET) overview. In *Proceedings of the TIPSTER Text Program: Phase II*, pages 445–447.
- Peter Mika. 2017. What happened to the semantic web? In *Proceedings of the 28th ACM Conference on Hypertext and Social Media*, pages 3–3.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.
- David Milne and Ian H. Witten. 2008. Learning to link with Wikipedia. In *Conference on Information and Knowledge Management*, pages 509–518.
- Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, volume 2, pages 1003–1011.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dhharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

- David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26.
- Ndapandula Nakashole, Tomasz Tylenda, and Gerhard Weikum. 2013. Fine-grained semantic typing of emerging entities. In *Annual Meeting of the Association for Computational Linguistics*, pages 1488–1497.
- Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 145–152.
- Dat Quoc Nguyen. 2017. An overview of embedding models of entities and relationships for knowledge base completion. *CoRR*, abs/1703.08098.
- Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48.
- Joel Nothman, James R. Curran, and Tara Murphy. 2008. Transforming wikipedia into named entity training data. In *Proceedings of the 2008 Australasian Language Technology Association Workshop*, pages 124–132.
- 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.
- Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206.
- L. Page, S. Brin, R. Motwani, and T. Winograd. 1998. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172.

- Rasmus Berg Palm, Dirk Hovy, Florian Laws, and Ole Winther. 2017. End-to-end information extraction without token-level supervision. In *Proceedings of the Workshop on Speech-Centric Natural Language Processing*, pages 48–52.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Annual Meeting on Association for Computational Linguistics*, pages 311–318.
- Rebecca J. Passonneau, Emily Chen, Weiwei Guo, and Dolores Perin. 2013. Automated pyramid scoring of summaries using distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–147.
- Máté Pataki, Miklós Vajna, and Attila Marosi. 2012. Wikipedia as text. *ERCIM News*, (89):48–49.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.
- Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *Annual Meeting of the Association for Computational Linguistics*, pages 183–190.
- Maria Pershina, Yifan He, and Ralph Grishman. 2015. Personalized page rank for named entity disambiguation. In *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 238–243.
- Matthew E. Peters and Dan Lecocq. 2013. Content extraction using diverse feature sets. In *Proceedings of the 22Nd International Conference on World Wide Web*, pages 89–90.

- Francesco Piccinno and Paolo Ferragina. 2014. From TagME to WAT: a new entity annotator. In *SIGIR Workshop on Entity Recognition and Disambiguation*, pages 55–62.
- Glen Alan Pink. 2017. *Slot filling*. Phd thesis, University of Sydney. Faculty of Engineering and Information Technologies. School of Information Technologies.
- Simone Paolo Ponzetto and Michael Strube. 2006. Exploiting semantic role labeling, WordNet and wikipedia for coreference resolution. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 192–199.
- Richard Power and Allan Third. 2010. Expressing OWL axioms by english sentences: Dubious in theory, feasible in practice. In *International Conference on Computational Linguistics*, pages 1006–1013.
- Will Radford. 2014. *Linking named entities to Wikipedia*. Ph.D. thesis, University of Sydney. Faculty of Engineering and Information Technologies. School of Information Technologies.
- 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 *Proceedings of the 2012 Text Analysis Conference*.
- 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 - Volume 1*, pages 814–824.
- Delip Rao, Paul McNamee, and Mark Dredze. 2010. Streaming cross document entity coreference resolution. In *Proceedings of the 23rd International Conference on Computational Linguistics: 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*, pages 147–155.

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 *Annual Meeting of the Association for Computational Linguistics*, pages 1375–1384.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 41–47.

Scott Reed, Zeynep Akata, Xinchun Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative adversarial text to image synthesis. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, pages 1060–1069.

Joseph Reisinger and Raymond J Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144.

- A.E. Richman and P. Schone. 2008. Mining Wiki Resources for Multilingual Named Entity Recognition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1–9.
- Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part III*, pages 148–163.
- Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84.
- Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, pages 474–479.
- Benjamin Roth, Tassilo Barth, Michael Wiegand, and Dietrich Klakow. 2013. A survey of noise reduction methods for distant supervision. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, pages 73–78.
- Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Conference on Empirical Methods in Natural Language Processing*, pages 379–389.
- Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.
- Barry Schiffman, Inderjeet Mani, and Kristian Concepcion. 2001. Producing biographical summaries: Combining linguistic knowledge with corpus statistics. In *Annual Meeting of the Association for Computational Linguistics*, pages 458–465.

- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.
- Wei Shan, Jiawei Han, and Jianyong Wang. 2014. A probabilistic model for linking named entities in web text with heterogeneous information networks. In *International Conference on Management of Data*, pages 1199–1210.
- W. Shen, J. Wang, and J. Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.
- Frank M. Shipman, III and Catherine C. Marshall. 1999. Formality considered harmful: Experiences, emerging themes, and directions on the use of formal representations in interactive systems. *Computer Supported Cooperative Work*, 8(4):333–352.
- Clay Shirky. 2010. *Cognitive surplus: Creativity and generosity in a connected age*. Allen Lane, London.
- Subramanian Shivashankar, Timothy Baldwin, Julian Brooke, and Trevor Cohn. 2017. Pairwise webpage coreference classification using distant supervision. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 841–842.
- 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.
- Ajit P. Singh and Geoffrey J. Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 650–658.
- 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 - Volume 1*, 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.
- Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934.
- Richard Socher, Cliff Chiung-Yu Lin, Andrew Y. Ng, and Christopher D. Manning. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 129–136.
- Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.*, 27(4):521–544.
- Sainbayar Sukhbaatar, Arthur Azlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28*, pages 2440–2448.
- Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Annual Conference on Neural Information Processing Systems*, pages 3104–3112.

Merine Thomas, Hiroko Bretz, Thomas Vacek, Ben Hachey, Sudhanshu Singh, and Frank Schilder. 2014. Newton: Building an authority-driven company tagging and resolution system. chapter 7. Chandos, Oxford, UK.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, pages 1–4.

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 Computational Natural Language Learning*, pages 142–147.

A. Toral and R. Munoz. 2006. A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia. In *Workshop on New Text, 11th Conference of the European Chapter of the Association for Computational Linguistics*.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394.

Olga Uryupina, Massimo Poesio, Claudio Giuliano, and Kateryna Tymoshenko. 2011. Disambiguation and filtering methods in using web knowledge for coreference resolution. In *Proceedings of the 24th International Florida Artificial Intelligence Research Society Conference*.

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 *Proceedings of the 2009 Text Analysis Conference*.

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 6000–6010.
- Patrick Verga, Emma Strubell, and Andrew McCallum. 2018. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.
- Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015a. Pointer networks. In *Advances in Neural Information Processing Systems 28*, pages 2692–2700.
- Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015b. Grammar as a foreign language. In *Annual Conference on Neural Information Processing Systems*, pages 2755–2763.
- Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. In *Proceedings of the International Conference on Machine Learning Deep Learning Workshop*.
- Pavlos Vougiouklis, Hady ElSahar, Lucie-Aimée Kaffee, Christophe Gravier, Frédérique Laforest, Jonathon S. Hare, and Elena Simperl. 2017. Neural wikipedia: Generating textual summaries from knowledge base triples. *CoRR*, abs/1711.00155.
- Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.
- 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.
- Claudia Wagner, David Garcia, Mohsen Jadidi, and Markus Strohmaier. 2015. It’s a man’s wikipedia? assessing gender inequality in an online encyclopedia. In *Proceedings of the Ninth International Conference on Web and Social Media, 2015*, pages 454–463.

- Richard C. Wang and William W. Cohen. 2007. Language-independent set expansion of named entities using the web. In *Proceedings of the 2007 International Conference on Data Mining*, pages 342–350.
- Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119.
- Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721.
- Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698.
- Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1366–1371.
- W. A. Woods. 1973. Progress in natural language understanding: An application to lunar geology. In *Proceedings of the June 4-8, 1973, National Computer Conference and Exposition, AFIPS '73*, pages 441–450.
- Fei Wu and Daniel S. Weld. 2007. Autonomously semantifying wikipedia. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, pages 41–50.
- Chunyang Xiao, Marc Dymetman, and Claire Gardent. 2016. Sequence-based structured prediction for semantic parsing. In *Annual Meeting of the Association for Computational Linguistics*, pages 1341–1350.

- Ying Xu, Zhiqiang Gao, Campbell Charles Wilson, Zhizheng Zhang, Man Zhu, and Qiu Ji. 2013. *Entity correspondence with second-order Markov logic*, pages 1 – 14. Springer Verlag, Germany.
- Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 250–259.
- Qing Yang, Peng Jiang, Chunxia Zhang, and Zhendong Niu. 2010. Reconstruct logical hierarchical sitemap for related entity finding. In *Proceedings of the 2010 Text Retrieval Conference*.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.
- Hamed Zamani and W. Bruce Croft. 2017. Relevance-based word embedding. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 505–514.
- Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.
- Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344.
- Lanbo Zhang, Yi Zhang, and Yunfei Chen. 2012. Summarizing highly structured documents for effective search interaction. In *International Conference on Research and Development in Information Retrieval*, pages 145–154.

- 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, Xiance Si, Fangtao Li, Edward Y. Chang, and Xiaoyan Zhu. 2012a. 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*, pages 82–89.
- Zhicheng Zheng, Xiance Si, Fangtao Li, Edward Y. Chang, and Xiaoyan Zhu. 2012b. Entity disambiguation with freebase. In *Proceedings of the 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 82–89.
- Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Volume 2: Short Papers*.