Artificial Intelligence 194 (2013) 130–150

Contents lists available at SciVerse ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Evaluating Entity Linking with Wikipedia

Ben Hachey [a,*], Will Radford [b,c], Joel Nothman [b,c], Matthew Honnibal [d], James R. Curran [b,c]

[a] Research & Development, Thomson Reuters Corporation, St. Paul, MN 55123, USA
[b] School of Information Technologies, University of Sydney, NSW 2006, Australia
[c] Capital Markets CRC, 55 Harrington Street, NSW 2000, Australia
[d] Department of Computing, Macquarie University, NSW 2109, Australia

ARTICLE INFO

Article history:
Available online 23 April 2012

Keywords:
Named Entity Linking
Disambiguation
Information extraction
Wikipedia
Semi-structured resources

ABSTRACT

Named Entity Linking (NEL) grounds entity mentions to their corresponding node in a Knowledge Base (KB). Recently, a number of systems have been proposed for linking entity mentions in text to Wikipedia pages. Such systems typically search for candidate entities and then disambiguate them, returning either the best candidate or NIL. However, comparison has focused on disambiguation accuracy, making it difficult to determine how search impacts performance. Furthermore, important approaches from the literature have not been systematically compared on standard data sets.

We reimplement three seminal NEL systems and present a detailed evaluation of search strategies. Our experiments find that coreference and acronym handling lead to substantial improvement, and search strategies account for much of the variation between systems. This is an interesting finding, because these aspects of the problem have often been neglected in the literature, which has focused largely on complex candidate ranking algorithms.

1. Introduction

References to entities such as people, places and organisations are difficult to track in text, because entities can be referred to by many mention strings, and the same mention string may be used to refer to multiple entities. For instance, David Murray might refer to either the jazz saxophonist or the Iron Maiden guitarist, who may be known by other aliases such as Mad Murray. These synonymy and ambiguity problems make it difficult for language processing systems to collect and exploit information about entities across documents without first linking the mentions to a knowledge base.

Named Entity Linking (NEL) is the task of resolving named entity mentions to entries in a structured Knowledge Base (KB). NEL is useful wherever it is necessary to compute with direct reference to people, places and organisations, rather than potentially ambiguous or redundant character strings. In the finance domain, NEL can be used to link textual information about companies to financial data, for example, news and share prices [34]. NEL can also be used in search, where results for named entity queries could include facts about an entity in addition to pages that talk about it [8].

NEL is similar to the widely-studied problem of word sense disambiguation (WSD, [36]), with Wikipedia articles playing the role of WordNet synsets [20]. At core, both tasks address problems of synonymy and ambiguity in natural language. The tasks differ in terms of candidate search and NIL detection. Search for WSD assumes that WordNet is a complete lexical resource and consists of a lexical lookup to find the possible synsets for a given word. The same approach is taken in wikification, where arbitrary phrases including names and general terms are matched to Wikipedia pages [32,33,27,15].

* Corresponding author.
E-mail address: ben.hachey@gmail.com (B. Hachey).

0004-3702/$ – see front matter
http://dx.doi.org/10.1016/j.artint.2012.04.005

However, this does not provide a mechanism for dealing with objects that are not present in the database. NEL, on the other hand, does not assume the KB is complete, requiring entity mentions without KB entries to be marked as NIL [8,31]. Furthermore, named entity mentions vary more than lexical mentions in WSD. Therefore, search for NEL requires a noisier candidate generation process, often using fuzzy matching to improve recall [48,28].

Until recently, wide-coverage NEL was not possible since there was no general purpose, publicly available collection of information about entities. However, Wikipedia has emerged as an important repository of semi-structured, collective knowledge about notable entities. Accordingly, it has been widely used for knowledge modelling [46,6,37,42]. It has been used for NLP tasks like automatic summarisation [45,50]. And it has also been exploited for a number of information extraction tasks ranging from NER learnt from Wikipedia link structure [40] to relation extraction learnt from the nearly structured information encoded in Wikipedia Infoboxes [51].

The most popular data sets for NEL were distributed as part of the recent Knowledge Base Population tasks at the NIST Text Analysis Conference (TAC). The thirteen participants in the 2009 task developed systems that linked a set of 3904 entity mentions in news and web text to a knowledge base extracted from Wikipedia infoboxes. The highest accuracy achieved was 82.2% [48] with subsequent publications reporting results as high as 86% [21].

The popularity of the TAC shared tasks has led to a wide range of innovative entity linking systems in the literature. However, since all participants were individually striving for the highest accuracy they could achieve, the systems all differ along multiple dimensions, so it is currently unclear which aspects of the systems are necessary for good performance and which aspects might be improved.

In this paper, we reimplement three prominent entity linking systems from the literature to obtain a better understanding of the named entity linking task. Our primary question concerns the relative importance of *search* and *disambiguation*: an NEL system must first search for a set of candidate entities that the mention string might refer to, before selecting a single candidate given the document. These phases have not been evaluated in isolation, and the systems from the literature tend to differ along both dimensions.

We find that the search phase is far more important than previously acknowledged. System descriptions have usually focused on complicated ranking methods. However, search accounts for most of the variation between systems. Furthermore, relatively unremarked search features such as query expansion based on coreference resolution and acronym detection seem to have a much larger impact on system performance than candidate ranking.

2. Review of named entity disambiguation tasks and data sets

Several research communities have addressed the named entity ambiguity problem. It has been framed in two different ways. Within computational linguistics, the problem was first conceptualised by Bagga and Baldwin [4] as an extension of the coreference resolution problem. Mihalcea and Csomai [32] later used Wikipedia as a word sense disambiguation data set by attempting to reproduce the links between pages, as link text is often ambiguous. Finally, Bunescu and Paşca [8] used Wikipedia in a similar way, but include NER as a preprocessing step and require a link or (NIL) for all identified mentions. We will follow the terminology of these papers, and refer to the three tasks respectively as *cross-document coreference resolution* (CDCR), *wikification*, and *named entity linking* (NEL). We use the more general term *named entity disambiguation* when we must avoid referring specifically to any single task.

The CDCR, wikification, and NEL tasks make different assumptions about the problem, and these lead to different evaluation measures and slightly different techniques. The CDCR task assumes that the documents are provided as a batch, and must be clustered according to which entities they mention. Systems are evaluated using clustering evaluation measures, such as the B^3 measure [3]. The wikification task assumes the existence of a knowledge base that has high coverage over the entities of interest, and that entities not covered by the knowledge base are relatively unimportant. And NEL requires a knowledge base but does not assume that it is complete. Systems are usually evaluated on micro-accuracy (percentage of *mentions* linked correctly) and macro-accuracy (percentage of *entities* linked correctly). In this section, we review the main data sets that have been used in CDCR and NEL research. Although we make some reference to approaches used, were reserve the main description of named entity disambiguation techniques for Section 3.

2.1. Early cross-document coreference datasets

The seminal work on cross-document coreference resolution (CDCR) was performed by Bagga and Baldwin [4]. They performed experiments on a set of 197 documents from the New York Times whose text matched the expression `John.*?Smith`—where `.*?` is a non-greedy wildcard match up to the first instance of Smith, e.g., only John Donnel Smith would be matched in John Donnell Smith bequeathed his herbarium to the Smithsonian. The documents were manually grouped according to which John Smith entities they mentioned. None of the articles mentioned multiple John Smiths, so the only annotations were at the document level.

The John Smith dataset approaches the problem as *one name, many people*: there are many entities that are referred to by an ambiguous name such as John Smith. However, there is another side to the problem: *one person, many names*. An entity known as John Smith might also be known as Jack Smith, Mr. Smith, etc. In other words, there are both synonymy and ambiguity issues for named entities.

Most CDCR datasets are similarly collected by searching for a set of canonical names, ignoring non-canonical coreferent forms. For instance, Mann and Yarowsky [29] collected a data set of web pages returned from 32 search engine queries for person names sampled from US census data. This data was later included in the WePS data described in Section 2.3. While ensuring that each document contains a canonical form for an ambiguous entity, this produces an unrealistic sample distribution.

In contrast, Day et al. [11] identify coreferent entity chains between documents in the ACE 2005 corpus [38], which already marks in-document coreference between proper name, nominal and pronominal entity mentions. Marking in-document and cross-document coreference for all entities in a corpus addresses both synonymy and ambiguity issues.

2.2. Generating data with pseudo-names

Because manually annotating data is costly, there has been some interest in adopting the *pseudo-words* strategy of generating artificial word sense disambiguation data first described by Gale et al. [16]. For word sense disambiguation, the data is generated by taking two words that are not sense ambiguous, and replacing all instances of them with an ambiguous key. For instance, all instances of the words banana and door would be replaced by the ambiguous key banana–door. The original, unambiguous version is reserved as the gold standard for training and evaluation.

Cross-document coreference resolved data can be generated in the same way by taking all instances of two or more names, and conflating them under an anonymisation key such as Person X. The task is then to group the documents according to their original name mentions. This strategy was first explored by Mann and Yarowsky [29], and subsequently by Niu et al. [39] and Gooi and Allan [17].

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

For named entity disambiguation, there is little reason to believe that two people named John Smith will share any more properties than one entity named Paul Simonell and another named Hugh Diamoni, so the criticism of pseudo-data that has been made about word sense disambiguation does not apply. On the other hand, named entities have interesting internal structures that a named entity disambiguation system might want to exploit. For instance, the use of a title such as Mr. or Dr. may be a critical clue. This makes named entities difficult to anonymise effectively under a key such as `Person X` without losing important information.

2.3. Web People Search

The first large data set for CDCR was distributed by the Web People Search shared task [1]. The data set consisted of up to 100 web search results for 49 personal names, for a total data set of 3489 documents manually sorted into 527 clusters. The task was repeated the following year, with a new evaluation set consisting of 3432 documents sorted into 559 clusters [2]. The most recent task, WePS-III, provided 57,956 documents from which the new evaluation data would be drawn—the top 200 search results for 300 person names. Only a subset of the documents received gold standard annotations.

WePS-III also added an additional entity disambiguation task, targeted at Online Reputation Management. The organisers searched the Twitter messaging service for posts about any of 100 companies, selected according to the ambiguity of their names—companies within names that were too ambiguous or too unambiguous were excluded. Mechanical Turk was used to cheaply determine which of 100 tweets per company name actually referred to the company of interest. Participants were supplied the tweets, the company name, and the URL of the company's homepage. This task is closer to named entity linking than cross-document coreference resolution, but shares a common weakness of CDCR data: the data was collected by searching for the company name, so the task does not address named entity synonymy.

2.4. Wikification

The development of Wikipedia offered a new way to approach the problem of entity ambiguity. Instead of clustering entities, as is done in CDCR, mentions could be resolved to encyclopedia pages. This was first described by Mihalcea and Csomai [32]. The task, which we refer to as *wikification*, is to add links from important concept mentions in text to the corresponding Wikipedia article. The task differs from Named Entity Linking in that concepts are not necessarily named entities, and in that the knowledge base is assumed to complete (i.e., presence in the encyclopedia is a minimum requirement for being identified and linked).

In order to encourage further research on wikification, the INEX workshops ran a Link the Wiki task between 2007 and 2009 [25]. The task is designed to improve Information Retrieval and places an emphasis on Wiki creation and maintenance as well as evaluation tools and methodologies. The 2009 task introduces a second wiki, Te Ara,[1] an expert-edited encyclopedia about New Zealand. Te Ara does not contain inter-article links, so the first subtask is to discover them. The second task is to link Te Ara articles to Wikipedia articles.

[1] http://www.teara.govt.nz/.

2.5. Named Entity Linking

The first attempts at what we term the *Named Entity Linking* (NEL) task—the task of linking entity mentions to a knowledge base—predicted the target of links in Wikipedia. This resembles the pseudo-name generation task described in Section 2.2, in that it makes a large volume of data immediately available, but the data may not be entirely representative. Cucerzan [9] has pointed out that the ambiguity of Wikipedia link anchor texts is much lower than named entity mentions in news data. This may be because the MediaWiki mark up requires editors to retrieve the article title in order to make a link, and they must then actively decide to use some other mention string to anchor the text. This seems to encourage them to refer to entities using more consistent terminology than writers of other types of text.

Bunescu and Paşca [8] were the first to use Wikipedia link data to train and evaluate a system for grounding text to a knowledge base. However, they did not evaluate their systems on manually linked mentions, or text from sources other than Wikipedia. The first to do so was Cucerzan [9], who evaluated on both Wikipedia and a manually linked set of 20 news articles, described in more detail in Section 2.7.

2.6. The Text Analysis Conference Knowledge Base Population challenge

The first large set of manually annotated named entity linking data was prepared by the National Institute of Standards and Technologies (NIST) as part of the Knowledge Base Population (KBP) shared task at the 2009 Text Analysis Conference (TAC) [31].

The 2009 TAC-KBP distributed a knowledge base extracted from a 2008 dump of Wikipedia and a test set of 3904 queries. Each query consisted of an ID that identified a document within a set of Reuters news articles, a mention string that occurred at least once within that document, and a node ID within the knowledge base. Little training data was provided.

Each knowledge base node contained the Wikipedia article title, Wikipedia article text, a predicted entity type (*per*, *org*, *loc* or *misc*), and a key-value list of information extracted from the article's infobox. Only articles with infoboxes that were predicted to correspond to a named entity were included in the knowledge base.

The annotators did not select mentions randomly. Instead, they favoured mentions that were likely to be ambiguous, in order to provide a more challenging evaluation. If the entity referred to did not occur in the knowledge base, it was labelled NIL. A high percentage of queries in the 2009 test set did not map to any nodes in the knowledge base—that is, the gold standard answer for 2229 of the 3904 queries was NIL.

The 2010 challenge used the same configuration as the 2009 challenge, and kept the same knowledge base. A training set of 1500 queries was provided, with a test set of 2250 queries. In the 2010 training set, only 28.4% of the queries were NIL, compared to 57.1% in the 2009 test data and 54.6% in the 2010 test data (details in Section 4 below). This mismatch between the training and test data may have harmed performance for some systems. Systems can be quite sensitive to the number of NIL queries, because it is difficult to determine whether a candidate that seems to weakly match the query should be discarded, in favour of guessing NIL. A high percentage of NIL queries thus favours conservative systems that stay close to the NIL baseline unless they are very confident of a match.

The most successful participants in the 2009 challenge addressed this issue by augmenting their knowledge base with articles from a recent Wikipedia dump. This allowed them to consider strong matches against articles that did not have any corresponding node in the knowledge base, and return NIL for these matches. This turned out to be preferable to assigning a general threshold of match strength below which NIL would be returned. We use the 30th July 2010 snapshot of English Wikipedia as a proxy KB for NEL. Since it is larger, it should provide more information to disambiguate candidate entities for mentions. After disambiguation, we then check to see if the linked entity exists in the KB, returning NIL for entities that we could link, but were not in the supplied KB.

2.7. Other NEL evaluation data

In addition to the data from the TAC challenge, three individual researchers have made their test sets available. Cucerzan [9] manually linked all entities from 20 MSNBC news articles to a 2006 Wikipedia dump, for a total of 756 links, with 127 resolving to NIL. This data set is particularly interesting because mentions were linked exhaustively over articles, unlike the TAC data, where mentions were selected for annotation if the annotators regarded them as interesting. The Cucerzan dataset thus gives a better indication of how a real-world system might perform.

Fader et al. [13] evaluate against 500 predicate–argument relations extracted by TextRunner from a corpus of 500 million Web pages, covering various topics and genres. Considering only relations where one argument was a proper noun, the authors manually identified the Wikipedia page corresponding to the first argument, assigning NIL if there is no corresponding page. 160 of the 500 mentions resolved to NIL.

Dredze et al. [12] performed manual annotation using a similar methodology to the TAC challenges, in order to generate additional training data. They linked 1496 mentions from news text to the TAC knowledge base, of which 270 resolved to NIL—a substantially lower percentage of NIL-linked queries than the 2009 and 2010 TAC data.

There is also some work on integrating linking annotation with existing NER datasets, including the CoNLL-03 English data [24] and ACE 2005 English data [5]. This is important since it allows evaluation of different steps of the pipeline of NErecognition, coreference (gold-standard in the latter case) and linking.

Table 1
Summary of named entity disambiguation data sets.

Task	Name	Year	Source	All mentions	Instances
CDCR	John Smith	1998	News	✘	197
CDCR	WePS 1	2007	Web	✘	3489
CDCR	Day et al.	2008	News	✔	3660
CDCR	WePS 2	2008	Web	✘	3432
CDCR	WePS 3	2009	Web	✘	31950
wikify	Mihalcea	2007	Wiki	✔	7286
wikify	Kulkarni	2009	Web	✔	17,200
wikify	Milne	2010	Wiki	✔	11,000
NEL	Cucerzan	2007	News	✔	797
NEL	TAC 09	2009	News	✘	3904
NEL	Fader	2009	News	✘	500
NEL	TAC 10	2010	News, Blogs	✘	3750
NEL	Dredze	2010	News	✘	1496
NEL	Bentivogli	2010	News, Web, Transcripts	✔	16,851
NEL	Hoffart	2011	News	✔	34,956

2.8. The BioCreative challenge Gene Normalisation task

The 2008 BioCreative workshop ran an entity linking challenge for biomedical text, which they termed Gene Normalisation (GN, [23,35]). Participants were provided the raw text of abstracts from scientific papers, and asked to extract the Entrez Gene identifiers for all human genes and proteins mentioned in the abstract. The GN task is motivated by genomics database curation, where scientific articles are linked to the genes/proteins of interest. The GN task differs from the real curation task in that it does not use the full text of the articles, and it annotates every human gene/protein mentioned (not just those described with new scientific results).

The version of the Entrez Gene database used for the task consists of a list of 32,975 human gene/protein identifiers, including an average of 5.5 synonyms each. Evaluation data was created by human experts trained in molecular biology and included 281 abstracts for training and 262 for testing. These sets have 684 and 785 total identifier annotations respectively, corresponding to averages of 2.4 and 3 per abstract. Inter-annotator agreement was reported as over 90%.

2.9. Database Record Linkage

Record Linkage [49] aims to merge entries from different databases, most commonly names and addresses for the same individual. This is often framed as database cleaning: canonical versions of names and addresses are produced, with duplicates sometimes removed in the process. Initial research by Fellegi and Sunter [14] presented a probabilistic description of the linkage problem and subsequent work extends this to use multiple sources of information or treats it as a graph of mentions to be partitioned into entity clusters. While similar to NEL, Record Linkage tends to consider more structured data (e.g., names and addresses) cleanly separated into database fields. This does, however, allow exploration of large datasets of person-related data (e.g., census and medical records), motivating work on efficiency and privacy.

2.10. Summary of Evaluation Sets

Table 1 shows the data sets used to evaluate named entity disambiguation work. Named entity disambiguation has been addressed as multiple tasks, including cross-document coreference resolution (CDCR), wikification (wikify), and named entity linking (NEL).

The CDCR data usually assumes that each document mentions one person of interest, usually using a canonical name form. The task is then to cluster the documents that refer to that person. In recent years, the task has been focused on the Web Person Search challenge datasets.

Named entity disambiguation is also sometimes addressed as part of wikification tasks. In these tasks, concepts must be identified and linked to the best Wikipedia page. Concepts are often named entities, but need not be. This is often evaluated on Wikipedia links directly, but Kulkarni et al. [27] point out that this leads to inaccurate performance estimates due to canonicalisation, so collected their own dataset of 17,200 terms mentions using web text from popular domains from a variety of genres.

Finally, NEL resembles wikification, but seeks to link all named entity mentions, requiring a mechanism for handling mentions that do not have a corresponding node in the knowledge base. Much of the work on this problem has been done using the TAC data sets. One weakness of these datasets is that they were collected by cherry-picking 'interesting' mentions, rather than systematically annotating all mentions within a document. One dataset that corrects this is described by Cucerzan [9]. However, the Cucerzan data was collected by correcting the output of his system, which may bias the data towards his approach. This may make the data unsuitable for comparison between systems.

Table 2
Comparative summary of seminal linkers.

System	Extractor	Searcher								Disambiguator
		Condition	Title	Redirect	Link	Truncated	Bold	DABTitle	Filter	
Bunescu and Paşca [8]	NER	NA	✔	✔				✔	NA	SVM rank over cosine and mention context word × category features
Cucerzan [9]	NER, coreference expansion	NA	✔	✔	✘	✔		✔	NA	Scalar product between candidate category/term vector and document-level vector
Varma et al. [48]	NER, acronym expansion	**if** acronym								Cosine between
		if expandable	✔						in KB	candidate
		else	✔	✔			✔	✔	NA	article term vector and
		else								
		search 1	✔						in KB	mention
		if no candidates	✔	✔			✔	✔	NA	context vector

3. Approaches

To date, the literature on named entity linking has largely consisted of detailed descriptions of novel complex systems. However, while NEL systems are commonly described in terms of separate search and disambiguation components,[2] very little analysis has been performed that looks at the individual effect of these components.

In this section, we describe our implementations of three such complex systems from the literature [8,9,48], in order to provide the first detailed analysis of the named entity linking task. These systems were selected for being seminal work on the task, for being highly novel, and for reporting very high performance. None of these systems have been compared against each other before.

3.1. A framework for Named Entity Linking

We suggest a Named Entity Linking (NEL) framework that allows replication and comparison of different approaches. The core task of an NEL system is to link a query mention, given its document context, to a Knowledge Base (KB) entity node or NIL. This can be separated into three main components: extractors, searchers and disambiguators.

Extractor. Extraction is the detection and preparation of named entity mentions. Most NEL datasets supply mention strings as queries. Some additional mention detection and preparation is often desirable however, because information about other entities in the text is useful for disambiguation. The extraction phase may also include other preprocessing such as tokenisation, sentence boundary detection, and in-document coreference. In-document coreference, in particular, is important as it can be used to find more specific search terms (e.g., *ABC* $\mapsto$ *Australian Broadcasting Corporation*).

Searcher. Search is the process of generating a set of candidate KB entities for a mention. Titles and other Wikipedia-derived aliases can be leveraged at this stage to capture synonyms (see Section 5 below). An ideal searcher should balance precision and recall to capture the *correct* entity while maintaining a small set of candidates. This reduces the computation required for disambiguation.

Disambiguator. In disambiguation, the best entity is selected for a mention. We frame this as ranking problem over the candidate set. We hold the NIL-detection strategy fixed for all disambiguators. This uses a Wikipedia snapshot from 30th July 2010 as a larger proxy KB for linking and any entities that do not exist in the small TACKB are returned as NIL.

Table 2 contains a summary of the extraction, search, and disambiguation components for our linker implementations, which are described in detail in the remainder of this section. Rows correspond to our implementations of seminal approaches from the literature. The first column for the searcher components contains conditions that need to be met for a given search to be performed. The following columns correspond to the alias sources used (see Section 5). And the last column specifies any filters that are applied to narrow the resulting candidate set.

[2] McCallum et al. [30] also describe a similar decomposition, motivated by efficiency, for the related task of clustering citation references.

3.2. Bunescu and Paşca

Bunescu and Paşca [8] were the first to explore the NEL task, using Support Vector Machines (SVM) to rank for disambiguation. However, its performance has not been compared against subsequent approaches.

Extractor. Bunescu and Paşca use data derived from Wikipedia for an evaluation whose goal is to return the correct target for a given link anchor, i.e., to re-introduce link targets in Wikipedia articles given the anchor text. They did not perform coreference or any other additional preprocessing.

Searcher. The search component for Bunescu and Paşca is an exact match lookup against article, redirect, and disambiguation title aliases. It returns all matching articles as candidates.

Disambiguator. The Bunescu and Paşca disambiguator uses a Support Vector Machine (SVM) ranking model, using the SVM^{light} toolkit [26]. Two types of features are used. The first feature type is the real-valued cosine similarity between the query context and the text of the candidate entity page (see Eq. (1) below). The second feature type is generated by creating a 2-tuple for each combination of candidate categories—Wikipedia classifications that are used to group pages on similar subjects—and context words. The categories are ancestors of those assigned to the candidate entity page, and the words are those that occurred within a 55-token context window of the entity mention. Based on results from Bunescu and Paşca, our implementation uses only categories that occur 200 times or more. However, while Bunescu and Paşca focused on `Person by occupation` pages in Wikipedia, the TAC data used for experiments here includes organisation and geopolitical entity types as well as a general person type (see Section 4 below). Thus, we explored general strategies for disambiguating arbitrary entity types. The union of great and great-great grandparent categories performed best in preliminary experiments and are used in our implementation here. Bunescu and Paşca include an NIL pseudo-candidate in the candidate list, allowing the SVM algorithm to learn to return NIL as the top-ranked option when no good candidate exists. We do not include NIL pseudo-candidates since this decreased performance in our development experiments (−0.5% *accuracy*). As mentioned above, this also allows us to hold the NIL-detection strategy constant for all disambiguation approaches. The learner is trained on the development data provided for the TAC 2010 shared task. It is important to note that the Bunescu and Paşca approach is the only one here that relies on supervised learning. The original paper derived training sets of 12,288 to 38,726 ambiguous person mentions from Wikipedia. Here, we use the TAC 2010 training data, which has 1500 total hand-annotated person, organisation, and geo-political entity mentions. The small size of this training set limits the performance of the machine learning approach in the experiments here. However, this also reflects the challenges of porting supervised approaches to different variations of the same task.

3.3. Cucerzan

Cucerzan [9] describes an NEL approach that focuses on an interesting document-level disambiguation approach. He also introduces a preprocessing module that identifies chains of coreferring entity mentions in order to use more specific name strings for querying. However, the effect of coreference handling on search and disambiguation is not explored.

Extractor. Cucerzan report an evaluation whose goal is to link all entity mentions in a news article to their corresponding Wikipedia page. Therefore, it is necessary to split the text into sentences, then detect and corefer named entity mentions. Cucerzan uses a hybrid NER tagger based on capitalisation rules, web and the CoNLL-03 NER shared task data [47] statistics. In our implementation, we first use the C&C NER tagger [10] to extract named entity mentions from the text. Next, naïve in-document coreference is performed by taking each mention and trying to match it to a longer, *canonical*, mention in the document. These are expected to be longer, more specific and easier to disambiguate. Mentions are examined in turn, longest to shortest, to see if it forms the prefix or suffix of a previous mention and is no more than three tokens shorter. Uppercase mentions are considered to be acronyms and mapped to a canonical mention if the acronym letters match the order of the initial characters of the mention's tokens. Our coreference implementation differs from that described by Cucerzan in that we do not require a canonical mention to have the same entity type as another mention coreferred to it, since we view identity as stronger evidence than predicted type.

Searcher. For candidate generation, canonical mentions are first case-normalised to comply with Wikipedia conventions. These are searched using exact-match lookup over article titles, redirect titles, apposition stripped article/redirect titles, and disambiguation titles. In contrast to Cucerzan, we do not use link anchor texts as search aliases because we found that they caused a substantial drop in performance (−5.2% KB *accuracy* on Cucerzan news data and approximately 10× worse runtime).

Disambiguator. Cucerzan disambiguated the query mention with respect to document-level vectors derived from all entity mentions. Vectors are constructed from the document and the global set of entity candidates, each candidate of each canonical mention. A *candidate* vector of indicator variables is created for each of the global candidates, based on presence of the article's categories and contexts. Contexts are anchor texts from the first paragraph or those that linked to another article

and back again. The extended document vector is populated to represent the union of indicator variables from all entity vectors. The category values are the number of entity vectors containing that category and the context values the count of that context in the document. Each candidate list for each mention is re-ranked separately with respect to the document-level vector. Specifically, candidates are ranked by the scalar product of the candidate vector and the extended document vector, with a penalty to avoid double-counting. Following Cucerzan, we exclude categories if their name contains any of the following words or their plurals: `article`, `page`, `date`, `year`, `birth`, `death`, `living`, `century`, `acronym`, `stub`; or a four-digit number (i.e., a year). We also exclude the `Exclude in print` category, which is used to mark content that should not be included in printed output. We do not shrink source document context where no clear entity candidate can be identified.

Benchmarking. We compared the performance of our reimplementation on the Cucerzan evaluation data (see Section 2.7), which consists of twenty news articles from MSNBC. This data includes 629 entity mentions that were automatically linked and manually verified by Cucerzan as linkable to Wikipedia articles. We achieved an accuracy of 88.3%, while Cucerzan reports an accuracy of 91.4%. There are several possible differences in our implementation. First, we are not certain whether we filter lists and categories using exactly the same heuristics as Cucerzan. We may also be performing coreference resolution, acronym detection or case-normalisation slightly differently. Changes in Wikipedia, especially the changes to the gold standard, may also be a factor. We observed that the evaluation was quite sensitive to small system variations, because the system tended to score either very well or rather poorly on each document, due to its global disambiguation model.

3.4. Varma et al.

Finally, Varma et al. [48] describe a system that uses a carefully constructed backoff approach to candidate generation and a simple text similarity approach to disambiguation. Despite the fact that it eschewed the complex disambiguation approaches of other submissions, this system achieved the best result (82.2% *accuracy*) at the TAC 2009 shared task.

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

Searcher. Different candidate generation strategies are followed for acronym and non-acronym queries. For *acronym queries*, if an expanded form of the query is found in the query document, then this is matched against KB titles. Otherwise, the original query string is used in an exact-match lookup against article/redirect/disambiguation titles, and bold terms in the first paragraph of an article. For *non-acronym queries*, the query string is first matched against KB titles. If no match is found, the query string is searched against the same aliases described above. The Varma et al. system for TAC 2009 also used metaphone search against KB titles for non-acronym queries. We omitted this feature from our implementation because Varma et al. reported that it degraded performance in experiments conducted after the TAC data was released (personal communication).

Disambiguator. The Varma et al. approach ranks candidates based on the textual similarity between the query context and the text of the candidate page, using the cosine measure. Here, the query context is the full paragraph surrounding the query mention, where paragraphs are easily identified by double-newline delimiters in the TAC source documents. The cosine score ranks candidates using the default formulation in Lucene:

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

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

4. Data

We report results on the TAC data sets. TAC queries consist of a mention string (e.g., Abbot) and a source document containing it (e.g., ... Also on DVD Oct. 28: "Abbot and Costello: The Complete Universal Pictures Collection"; ...). The gold standard is a reference to a TAC KB node (e.g., `E0064214`, or `Bud Abbott`), or NIL if there is no corresponding node in the KB. TAC source documents are drawn from newswire and blog collections. We extract and store body text, discarding markup and non-visible content if they are formatted using a markup language. After tokenising, we defer any further processing to specific extractors.

Table 3
Comparison of TAC data sets for all queries ($\mathcal{Q}$) and for unique entities ($\mathcal{E}$).

$\lvert\mathcal{Q}\rvert$	TAC 2009 test		TAC 2010 train		TAC 2010 test	
	3904		1500		2250	
KB	1675	(43%)	1074	(72%)	1020	(45%)
NIL	2229	(57%)	426	(28%)	1230	(55%)
PER	627	(16%)	500	(33%)	751	(33%)
ORG	2710	(69%)	500	(33%)	750	(33%)
GPE	567	(15%)	500	(33%)	749	(33%)
News	3904	(100%)	783	(52%)	1500	(67%)
Web	0	(0%)	717	(48%)	750	(33%)
Acronym	827	(21%)	173	(12%)	347	(15%)
$\lvert\mathcal{E}\rvert$	560		–		871	
KB	182	(33%)	462	(–)	402	(46%)
NIL	378	(67%)	–	(–)	469	(54%)
PER	136	(24%)	–	(–)	334	(38%)
ORG	364	(65%)	–	(–)	332	(38%)
GPE	60	(11%)	–	(–)	205	(24%)

The TAC KB is derived from pages in the October 2008 Wikipedia dump[3] that have infoboxes. It includes approximately 200,000 PER nodes, 200,000 GPE nodes, 60,000 ORG nodes and more than 300,000 miscellaneous/non-entity nodes. We also exploit a more recent English Wikipedia dump from 30th July 2010. This consumes 11.8 GB on disk with `bzip2` compression, including markup for 3.3 M articles. We use the `mwlib`[4] Python package to extract article text, categories, links, disambiguation and redirect information, and store them using Tokyo Tyrant,[5] a fast database server for Tokyo Cabinet key-value stores. This provides fast access to article data structures by title as well as the ability to stream through all articles.

We use the TAC 2009 test data as our main development set, so that we can benchmark against a large set of published results. We use the TAC 2010 training data for training the Bunescu and Paşca [8] disambiguator. And we reserve the TAC 2010 test data as our final held-out test set. These are summarised for all queries in the top part of Table 3. The first thing to note is the difference in the proportion of NIL queries across data sets. In both the TAC 2009 and TAC 2010 test sets, it is approximately 55%. However, in the TAC 2010 training set, it is considerably lower at 28%. The second difference is in the distribution of entity types. The TAC 2009 test data is highly skewed towards ORG entities while the TAC 2010 training and test data sets are uniformly distributed across PER, ORG and GPE entities. Finally, while TAC 2009 consisted solely of newswire documents, TAC 2010 included blogs as well. The TAC 2010 training data is roughly evenly divided between news and web documents (blogs), while the test data is skewed towards news (67%).

The bottom part of Table 3 contains the corresponding numbers (where defined) for unique entities. Note that this analysis is not possible for the TAC 2010 training data, since its NIL queries have not been clustered. The main difference between the data sets is in terms of the average number of queries per entity ($\lvert\mathcal{Q}\rvert/\lvert\mathcal{E}\rvert$)—7 for TAC 2009 compared to 2.6 for TAC 2010 test. The proportion of NIL queries is the same as in the query-level analysis at approximately 55% for the TAC 2009 and 2010 test sets. The distribution across entity types is similarly skewed for the TAC 2009 data. Where the query-level analysis for the TAC 2010 test data showed a uniform distribution across entity types, however, the entity-level analysis shows a substantial drop in the proportion of GPE entities.

4.1. Evaluation measures

We use the following evaluation measures, defined using the notation in Table 4. The first, *accuracy* (A), is the official TAC measure for evaluation of end-to-end systems. TAC also reports KB *accuracy* ($A_{\mathcal{C}}$) and NIL *accuracy* ($A_{\emptyset}$), which are equivalent to our candidate recall and NIL recall with a maximum candidate set size of one. The remaining measures are introduced here to analyse candidate sets generated by different search strategies.

accuracy **(A)**: percentage of correctly linked queries.

$$A = \frac{|\{C_{i,0} | C_{i,0} = \mathcal{G}\}|}{N} \tag{2}$$

[3] http://download.wikimedia.org.
[4] http://code.pediapress.com/wiki/wiki/mwlib.
[5] http://fallabs.com/tokyotyrant/.

Table 4
Notation for searcher analysis measures.

N	Number of queries in data set
$\mathcal{G}$	Gold standard annotations for data set ($\|\mathcal{G}\| = N$)
$\mathcal{G}_i$	Gold standard for query i (KB ID or NIL)
$\mathcal{C}$	Candidate sets from system output ($\|\mathcal{C}\| = N$)
$\mathcal{C}_i$	Candidate set for query i
$\mathcal{C}_{i,j}$	Candidate at rank j for query i (where $\mathcal{C}_i \neq \emptyset$)

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

$$\langle C \rangle = \frac{\sum_i |\mathcal{C}_i|}{N} \tag{3}$$

candidate precision ($P_{\mathcal{C}}$): percentage of non-empty candidate sets containing the correct entity.

$$P_{\mathcal{C}} = \frac{|\{\mathcal{C}_i | \mathcal{C}_i \neq \emptyset \wedge \mathcal{G}_i \in \mathcal{C}_i\}|}{|\{\mathcal{C}_i | \mathcal{C}_i \neq \emptyset\}|} \tag{4}$$

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

$$R_{\mathcal{C}} = \frac{|\{\mathcal{C}_i | \mathcal{G}_i \neq \text{NIL} \wedge \mathcal{G}_i \in \mathcal{C}_i\}|}{|\{\mathcal{G}_i | \mathcal{G}_i \neq \text{NIL}\}|} \tag{5}$$

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

$$P_{\emptyset} = \frac{|\{\mathcal{C}_i | \mathcal{C}_i = \emptyset \wedge \mathcal{G}_i = \text{NIL}\}|}{|\{\mathcal{C}_i | \mathcal{C}_i = \emptyset\}|} \tag{6}$$

NIL ***recall*** ($R_{\emptyset}$): percentage of NIL queries for which the candidate set is empty. A high $R_{\emptyset}$ rate is valuable because it is difficult for disambiguators to determine whether queries are NIL-linked when candidates are returned.

$$R_{\emptyset} = \frac{|\{\mathcal{C}_i | \mathcal{G}_i = \text{NIL} \wedge \mathcal{C}_i = \emptyset\}|}{|\{\mathcal{G}_i | \mathcal{G}_i = \text{NIL}\}|} \tag{7}$$

5. Wikipedia alias extraction

We extract a set of aliases—potential mention strings that can refer to an entity—for each Wikipedia article. By querying an index over these aliases, we are able to find candidate referents for each entity mention. We consider the following attributes of an article as candidate aliases:

Article titles (Title) The canonical title of the article. While the first character of Wikipedia titles is case-insensitive and canonically given in the uppercase form, for articles containing the special `lowercase title` template (such as `gzip`, `iPod`), we extract this alias with its first character lowercased.

Redirect titles (Redirect) Wikipedia provides a redirect mechanism to automatically forward a user from non-canonical titles—such as variant or erroneous spellings, abbreviations, foreign language titles, closely-related topics, etc.—to the relevant article. For articles with `lowercase title`, if the redirect title begins with the first word of the canonical title, its first character is also lowercased (e.g., IPods becomes iPods).

Bold first paragraph terms (Bold) Common and canonical names for a topic are conventionally listed in bold in the article's first paragraph.

Link anchor texts (Link) Links between Wikipedia articles may use arbitrary anchor text. Link anchors offer a variety of forms used to refer to the mention in running text, but the varied reasons for authors linking makes them noisy. We therefore extract all anchor texts that have been used to link to the article at least twice.

Disambiguation page titles (DABTitle) Disambiguation pages are intended to list the articles that may be referred to by an ambiguous title. The title of a disambiguation page (e.g., a surname or an abbreviation) is therefore taken as an alias of the pages it disambiguates.

Disambiguation pages usually consist of one or more lists, with each list item linking to a candidate referent of the disambiguated term. However, such links are not confined exclusively to candidates; based on our observations, we only consider links that appear at the beginning of a list item, or following a single token (often a determiner). All descendants of the `Disambiguation pages` category are considered disambiguation pages.

Disambiguation redirects and bold text (DABRedirect) One page may disambiguate multiple terms—for instance, there is one disambiguation page for both Amp and AMP. In addition to the page title, we therefore also consider bold terms in the page and the titles of redirects that point to disambiguation pages as aliases of the articles they disambiguate.

Table 5
Sources of aliases, including the number of articles (excluding disambiguation pages) and aliases with each source. *Support* indicates the average number of sources that support an alias.

Source	# Articles	# Aliases		Support
		without truncation	with truncation	
Article title	3 198 290	3 198 290	3 777 818	3.4
Redirect title	1 493 931	3 960 765	4 393 709	1.8
Bold terms	2 984 381	3 601 296	3 601 296	2.8
Link anchor	2 728 066	5 320 423	5 320 423	2.5
Disamb. title	933 308	1 126 714	1 203 648	3.7
Disamb. redirect	907 330	1 312 327	1 312 327	3.3
Disamb. bold	536 438	1 563 650	1 650 858	2.3
Disamb. hatnote	90 564	96 649	115 524	2.8
Any	3 198 290		17 156 466	1.6

Table 6
Search over individual alias fields (TAC 2009).

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

Disambiguation hatnotes (Hatnote) Even when a name or other term is highly ambiguous, one of the referents is often far more frequently intended than the others. For instance, there are many notable people named John Williams, but the composer is far more famous than the others. At the top of such an article, a link known as a *hatnote template* points to disambiguation pages or alternative referents of the term. We extract disambiguation information from many of the hatnote templates in English Wikipedia, and use the referring article's title as an alias, or the disambiguated redirect title specified in the template.

Truncated titles (Truncated) Wikipedia conventionally appends disambiguating phrases to form a unique article title, as in `John Howard (Australian actor)` or `Sydney, Nova Scotia`. For all alias sources that are titles or redirects, we strip expressions in parenthesis or following a comma from the title, and use the truncated title as an additional alias.

We store the alias sources as features of each article-alias pair, and use them to discriminate between aliases in terms of reliability. Titles and redirects are unique references to an article and are therefore considered most reliable, while link texts may require context to be understood as a reference to a particular entity. Table 5 indicates that while aliases derived from link texts are numerous, they are much less frequently supported by other alias sources than are disambiguation page titles.

The extracted aliases are indexed using the Lucene[6] search engine. Aliases are stored in Lucene *keyword* fields which support exact match lookup. We also index the Wikipedia text. Article text is stored in Lucene *text* fields which are used for scoring matches based on terms from entity mention contexts in source documents. The entire index occupies 12 GB of disk space, though this includes all the fields required for our experiments. Note that all experiments reported here set the Lucene query limit to return a maximum of 1000 candidates.

5.1. Coverage of alias sources

Table 6 shows the *candidate count*, *candidate recall*, *candidate precision*, NIL *recall* and NIL *precision* for the different alias sources used on our development set, TAC 2009. The first thing to note is the performance of the Title alias source. Title queries return 0 or 1 entities, depending on whether there was an article whose title directly matched the query. The *candidate count* of 0.2 indicates that 20% of the query mentions matched Wikipedia titles. These matches return the correct entity for 37.2% of the non-NIL queries. Precision over these title-matched non-NIL queries was 83.5%. This means that systems may benefit from a simple heuristic that trusts direct title matches, and simply returns the entity if a match is found.

[6] http://lucene.apache.org/.

Table 7
Search over multiple alias fields (TAC 2009).

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

Table 8
Backoff search over alias fields (TAC 2009).

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

It is very rare for a direct title match to be returned when the answer is actually NIL: this only occurred for 3.5% of the queries. It was, however, common for title match failures to occur for non-NIL queries. This can be seen in the NIL *precision* figure, which is only 68.1%. A title-match system that returns an entity whose title matches the query, or NIL otherwise, achieves 71.0% *accuracy* on the end-to-end linking task (TAC 2009). This is a fairly strong baseline—half of the 35 runs submitted to TAC 2009 scored below it. Expanding this system to also consult redirect titles improves this baseline to 76.3% linking *accuracy*. Only 5 of the 14 TAC 2009 teams achieved higher accuracy than this. The other alias sources potentially return multiple candidates, so their utility depends on the strength of the disambiguation component.

Table 7 shows how the number of candidates proposed increases as extra alias sources are considered, and how much *candidate recall* improves. The addition of link anchor texts increases *candidate recall* to 81.7%, but also greatly increases the number of candidates suggested. The NIL *recall* drops from 92.6% to 59.4%, which means that at least one candidate has been proposed for over 40% of the NIL-linked queries. This makes some form of NIL detection necessary, either through a similarity threshold, or a supervised model, as used by Zheng et al. [54]. Using all alias sources produces a *candidate recall* of 87.8%, with a mean of 7.2 candidates returned per query. The *candidate recall* constitutes an upper bound on linking KB *accuracy*. That is, there are 12.2% of KB-linked queries which even a perfect disambiguator would not be able to answer correctly. Many of these queries are acronyms or short forms that could be retrieved by expanding the query with an appropriate full-form from the source document (see experiments and analysis in Sections 6.2, 7, 8.2, and 9 below).

One way to reduce the number of candidates proposed is to use a *backoff* strategy for candidate generation. Using this strategy, the most reliable alias sources are considered first, and the system only consults the other alias sources if 0 candidates are returned. Table 8 shows the performance of the backoff strategy as each alias source is considered, ordered according to their *candidate precision*. A maximum of 2.4 candidates is returned, with a *candidate recall* of 77.1%. This may be a good strategy if a simple disambiguation system is employed, such as cosine similarity.

6. Analysis of searcher performance

Having described our reimplementations of several named entity linking systems, we now examine their performance in more detail, beginning with the accuracy of their *searchers*—that is, how accurately the systems propose candidates from mention strings.

6.1. Comparison of implemented searchers

Table 9 contains analysis results for our searcher reimplementations. The first row describes the performance of our Bunescu and Paşca searcher, which uses exact match over article, redirect, and disambiguation title aliases. The second row describes our Cucerzan searcher, which includes coreference and acronym handling. As described in Section 3.3, mentions are replaced by full-forms, as determined by coreference and acronym detection heuristics. The query terms are searched using exact match over article, redirect, and disambiguation titles, as well as apposition-stripped article and redirect titles. Finally, the third row describes our Varma et al. searcher, which replaces acronyms with full-forms where possible and employs a backoff search strategy that favours high-precision matching against article titles that map to the KB over alias

Table 9
Performance of searchers from the literature (TAC 2009).

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

Table 10
Effect of coreference/acronym handling on searcher performance (TAC 2009).

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

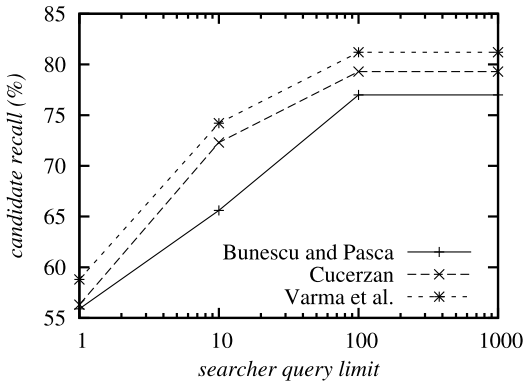

Fig. 1. Effect of query limit on searcher candidate recall.

search. Alias search includes exact match over article, redirect, and disambiguation titles, as well as bold terms in the first paragraph of an article.

The implemented Cucerzan and Varma et al. perform best. They both achieve *candidate precision* of close to 60% at candidate recall near 80%. This suggests that coreference and acronym handling are important and that a preference for high-precision matching is also beneficial. The Varma et al. searcher is slightly better in terms of *candidate precision* (+1.2%) and *candidate recall* (+1.9%). It also returns a candidate set size that, on average, contains 0.2 fewer items. This corresponds to a reduction in ambiguity of 6.3% with respect to the Cucerzan searcher.

6.2. Effect of extractors on search

Table 10 contains a subtractive analysis of coreference and acronym handling in searchers from the literature. The respective components result in less ambiguity (−0.9 for Cucerzan and −0.8 for Varma et al.) and a simultaneous increase in *candidate precision* (+5.2% and +5.8 respectively). For Varma et al., there is also an increase in *candidate recall* (+1.8%). This highlights the importance of using more specific mention forms where possible, as they are more likely to match the canonical names that occur in Wikipedia.

6.3. Effect of query limit on searcher candidate recall

One way to improve disambiguation efficiency is to reduce the number of candidates that must be considered. However, the correct candidate is not always the first one returned by the searcher. Fig. 1 plots the *candidate recall* of our searcher implementations against the query limit—the maximum number of results returned by the lucene alias index. All three linkers start with *candidate recall* under 60% and climb to their maximum at a query limit of 1000. Interestingly, there appears to be a knee at 100 for all three searchers, which suggests the possibility of some efficiency gain. However, going from a query limit of 100 down to 10 results in a substantial drop in *candidate recall*, especially for the Bunescu and Paşca searcher. Despite the possible efficiency gain, for the remaining experiments here we keep the query limit at 1000 so that our implementations are as close as possible to the literature.

Table 11
Number of KB *accuracy* errors due to search (TAC 2009).

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

Table 12
Distribution of searcher errors on TAC 2009 queries.

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

Table 13
Coreference analysis over 100 queries sampled from the TAC 2009 queries.

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

7. Searcher errors

In this section, we investigate the types of errors made by each of the three systems we implemented. The first question we asked was whether systems were making errors because their searchers were failing to find the candidates. Table 11 shows the number of search errors for each system. It also shows the total number of linking KB *accuracy* errors (due to either searchers or disambiguators) in the third column. The last row shows the number of queries for which all three systems returned an incorrect result. On average, 43% of KB *accuracy* errors are due to search recall problems. It is also interesting to note that a large proportion of the searcher error queries were common to all systems.

Table 12 shows the distribution of the common search errors, classified into broad categories. The Type column contains error totals over unique query mention strings, while the Token column contains error totals over individual queries. The most common type of search error occurs when a mention is underspecified or ambiguous (e.g., Health Department). Name variations—including nicknames (e.g., Cheli for Chris Chelios), acronyms (e.g., ABC), transliterations (e.g., Air Macao instead of Air Macau), and inserted or deleted tokens (e.g., Ali Akbar Khamenei instead of Ali Khamenei)—are also problematic. There are a few cases that may indicate annotation errors. For example, several gold standard articles are disambiguation pages, or have existed since before the dataset was prepared. Other errors are due to targeting a mention at an incorrect point in an organisational structure. The distinction between general university sports teams and the teams for baseball, for example, is subtle and proved very difficult for the systems to draw. There are also some legitimate typographic errors: Blufton should be Bluffton.

We also investigated the impact of coreference on linking performance over a sample of 100 queries drawn at random from the TAC 2009 data. Table 13 contains the counts of these queries that can be coreferred to a more specific mention and the count that are acronyms. Among the 24 coreferrable queries, our Cucerzan coreference module correctly resolves 5 and our Varma et al. acronym expansion module correctly resolves 6—three in common. Both systems correctly corefer some acronyms, including DCR ↦ Danish Council for Refugees, DMC ↦ DeLorean Motor Co. The Varma et al. coreference additionally corefers more acronym cases such as CPN-UML ↦ Communist Party of Nepal (Unified Marxist-Leninist) and TSX ↦ Tokyo Stock Exchange. Since the Cucerzan implementation only corefers NES, NE boundary detection error can rule out coreferring some acronyms, but correctly handles Cowboys ↦ Dallas Cowboys and Detroit ↦ Detroit Pistons. Note that while most acronyms are coreferrable, only half of the coreferrable queries are acronyms, indicating that coreference is advantageous but risks introducing complexity and potentially error.

8. Analysis of disambiguator performance

Next, we examine disambiguator performance in more detail, beginning with the end-to-end accuracy of implemented linkers.

Table 14
Comparison of systems from the literature (TAC 2009).

System	A	A_C	$A_\emptyset$
NIL baseline	57.1	0.0	100.0
Title baseline	71.0	37.2	96.5
+Redirect baseline	76.3	54.6	92.6
Bunescu and Paşca	77.0	67.8	83.8
Cucerzan	78.3	71.3	83.5
Varma et al. replicated	80.1	72.3	86.0
TAC 09 Median	71.1	63.5	78.9
TAC 09 Max (Varma)	82.2	76.5	86.4

Table 15
Effect of coreference/acronym handling on end-to-end linking performance (TAC 2009).

System	A	A_C	$A_\emptyset$
Cucerzan	78.3	71.3	83.5
− coreference handling	74.9	69.4	79.0
Varma et al.	80.1	72.3	86.0
− acronym handling	77.3	69.7	83.0

8.1. Comparison of implemented linkers

Table 14 summarises the performances of the different systems on the TAC 2009 test data. In addition to the systems described above, we report an NIL baseline that returns NIL for every query. Thus the overall accuracy of 57.1% reflects the number of NIL queries in the data set. We also report baselines based on exact matching against Wikipedia article titles, and exact matching against article titles and redirect titles (Section 5.1). The Title + Redirect baseline in particular is a strong baseline for this task, achieving a score 5.2 points above the median and 5.9 points below the maximum score achieved by submissions to the shared task. The last two rows correspond to the median and maximum results from the TAC 2009 proceedings, where the maximum corresponds to the reported results from Varma et al.

Of the systems we implemented, the Varma et al. approach performs best on this data, followed by Cucerzan. The Cucerzan and the Bunescu and Paşca systems perform only slightly better than the Title + Redirect baseline system, which does not use any disambiguation, and simply queries for exact matches for the mention string over the title and redirect fields. However, both systems would have placed just outside the top 5 at TAC 2009.

While the Varma et al. system was the best system submitted to TAC 2009, two recent papers have reported higher scores on the same data. Zheng et al. [54] report an *accuracy* of 84.9%, the highest in the literature, using an approach based on learnt ranking with ListNet and a separate SVM classifier for NIL detection over a diverse feature set. Zhang et al. [53] report an *accuracy* of 83.8%, using a classifier for NIL detection built over a large training set derived from Wikipedia. Nevertheless, the competitiveness of the Varma et al. approach still suggests that a good search strategy is critical to NEL, while different disambiguators have much less impact.

8.2. Effect of extractors on disambiguation

Table 15 contains a subtractive analysis of coreference and acronym handling in disambiguators from the literature. In Table 10 above (effect of extractors on search), we saw that this resulted in lower ambiguity without significantly affecting precision or recall. Here, we see that this results in substantial improvements in *accuracy* (A) of approximately 3 points. For our Cucerzan implementation, the difference is mainly in terms of NIL *accuracy*, which sees a 4.5 point increase due to the use of more specific name variants for search. Our Varma et al. implementation sees a more balanced increase in KB *accuracy* and NIL *accuracy* of approximately 3 points each. The relatively large increase in KB *accuracy* for Varma et al. may be due to its search of the entire document for acronym expansions, rather than just other entity mentions as is the case for our Cucerzan coreference handling. This makes the acronym expansion less vulnerable to Named Entity Recognition errors.

We also evaluated linker performance over the 100 query sample mentioned in Section 7 above. On this sample, adding coreference/acronym handling allowed our Cucerzan and Varma et al. implementations to correctly link one more query each.

8.3. Effect of searchers on disambiguation

Table 16 contains results for versions of our Bunescu and Paşca and Cucerzan implementations that use the described candidate search strategies, but replace the disambiguation approach with the simple cosine disambiguator described in Section 3.4. The results here relate directly to the search results in Table 9 (comparison of implemented searchers), with high *accuracy* achieved by the searchers that have high *candidate recall* and low *candidate count*. In Table 9, the Varma et al.

Table 16
Effect of searchers on cosine disambiguation (TAC 2009).

Searcher	A	$A_{\mathcal{C}}$	$A_{\emptyset}$
Bunescu and Paşca	77.7	69.6	83.8
Cucerzan	78.8	69.7	85.6
Varma et al.	80.1	72.3	86.0

Table 17
Combinations of searchers on implemented disambiguators (TAC 2009).

Searcher	Disambiguator	A	$A_{\mathcal{C}}$	$A_{\emptyset}$
Bunescu and Paşca	Bunescu and Paşca	77.0	69.6	83.8
Varma et al.	Bunescu and Paşca	78.1	67.9	85.8
Cucerzan	Cucerzan	78.3	71.3	83.5
Varma et al.	Cucerzan	79.4	73.3	83.9

Table 18
Number of KB *accuracy* errors due to disambiguation.

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

searcher outperforms the Bunescu and Cucerzan searchers in terms of *candidate recall* by 1.9 and 4.2 points respectively, and in terms of *candidate count*by 0.2 and 0.6. Here, it also performs best in terms of *accuracy* at 80.1%—2.4 points better than Bunescu and 1.3 point better than Cucerzan.

Note that the Bunescu and Paşca and Cucerzan disambiguators (Table 14) perform worse than the cosine disambiguators reported here. This may be attributed in part to differences between the training and development testing data. For example, the distributions between NIL and KB queries changes as described above in Table 3. Also, the TAC 2010 training data includes web documents while the TAC 2009 evaluation data used for development testing here does not. For Bunescu and Paşca, the difference may also be due in part to the fact that the training data is fairly small. The held-out evaluation data used in Section 10 is more similar to the training data. Results on this data (Table 21 below) suggest that the Bunescu and Paşca learning-to-rank disambiguator obtains higher *accuracy* than the corresponding cosine disambiguator (+0.7%), with a 1.5 point increase in *candidate recall*.

8.4. Effect of swapping searchers

Table 17 contains a comparison of the Bunescu and Paşca and the Cucerzan disambiguators using the search strategy they describe and the search strategy from Varma et al.[7] For the Cucerzan system, we use Varma et al. search for the TAC query only and Cucerzan search for the other named entity mentions in the document. The results suggest that the high-precision Varma et al. search is generally beneficial, resulting in an increase in *accuracy* (+1.1%) for both the Bunescu and Paşca and the Cucerzan disambiguators. Both of these results suggest that selecting a good search strategy is crucial.

9. Disambiguator errors

Table 18 shows the number of disambiguator errors—queries in the TAC 2009 data where the correct link was not returned because the disambiguator was unable to choose the correct candidate from the search results. It also shows the total number of KB *accuracy* errors (due to either searchers or disambiguators). The last row shows the number of queries for which all three systems return an incorrect result. The errors here account for the remaining errors (approximately 47%) that were not attributed to the searchers in Table 11 above. Interestingly, where search errors were largely common to all systems, few disambiguation errors are shared. Given the variation in performance and diversity of errors among the systems compared here, it is tempting to explore voting. However, many of the approaches described here already require substantial resources for large-scale applications (e.g., linking all mentions in a news archive containing decades worth of articles). We believe it is more important to explore efficiency improvements in future work. Therefore, we do not report voting experiments here.

[7] Note that the Varma et al. disambiguator corresponds to our cosine disambiguator. Therefore, the cosine disambiguation rows in Tables 14 and 21 correspond to the Bunescu and Paşca and Cucerzan systems with Varma et al. disambiguation. Note also that we do not swap in the Bunescu and Paşca searcher since it is not competitive (as discussed in Section 6.1).

Table 19
Distribution of disambiguator errors on TAC 2009 queries.

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

Table 20
Characteristic errors over TAC 2009 queries.

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

Table 21
Comparison of systems from the literature on the TAC 2010 test data.

System	A	$A_{\mathcal{C}}$	$A_{\emptyset}$
NIL baseline	54.7	0.0	100.0
Title baseline	69.6	35.0	98.4
+Redirect baseline	79.4	60.6	95.0
Bunescu and Paşca (CosDAB)	80.1	67.1	90.9
Cucerzan (CosDAB)	81.0	71.1	89.3
Bunescu and Paşca	80.8	68.4	91.1
Cucerzan	84.5	78.4	89.5
Varma et al.	81.6	70.5	90.7
TAC 2010 Median	68.4	–	–
TAC 2010 Maximum (Lehmann)	86.8	80.6	92.0

Table 19 shows a breakdown of the common errors. The types of errors are less varied than search errors, and are dominated by cases where the entities have similar names and are from similar domains. Name variation still makes up a reasonable proportion of the errors at this stage, but these are exclusively acronyms (i.e., there are no nicknames, transliterations, or insertions/deletions as in the search errors above).

Finally, Table 20 summarises the counts of queries for which each system returned an incorrect entity while the other two did not. The errors are categorised according to whether the mention was an acronym or not, and counts are aggregated at type and token granularity. The relative proportion of acronym and non-acronym errors differs slightly for the three systems, with Bunescu and Paşca making more acronym errors, while Cucerzan balances the two, and Varma et al. makes more errors on non-acronyms. This trend reflects the level of acronym processing: Bunescu and Paşca has none whereas Varma et al. uses a finely tuned acronym search and Cucerzan uses coreference. The counts over tokens broadly follow the same trend, although skewed by the bursty distribution of types and tokens.

10. Final results

As a final comparison, we evaluate our implementations of seminal systems on the TAC 2010 test data, which we set aside during system development. The results are shown in Table 21. Results columns correspond to the official TAC evaluation measures, which include *accuracy* (A), KB *accuracy* ($A_{\mathcal{C}}$) and NIL *accuracy* ($A_{\emptyset}$). Rows correspond to systems. The NIL baseline is a system that returns NIL for every query. The overall accuracy of 54.7% here reflects the percentage of queries with NIL as the gold answer. The Title baseline system performs an exact match lookup on Wikipedia titles. The Title + Redirect baseline performs an exact match on the union of article and redirect titles. The next three rows correspond to our implementations of the Bunescu and Paşca, Cucerzan, and Varma et al. systems.

Finally, the last two rows contain the median and maximum system scores from TAC 2010. The maximum was obtained by Lehmann et al. [28], whose searcher differs from those explored here in using token-based (rather than exact-match) search, coreference filtering, and Google search. The Lehmann et al. disambiguator uses features based on alias trustworthiness, mention-candidate name similarity, mention-candidate entity type matching, and Wikipedia citation overlap between candidates and unambiguous entities from the mention context. A heuristic over the features is used for candidate ranking. And a supervised binary logistic classifier is used for NIL detection.

The Cucerzan system is the most accurate of our systems on the evaluation data, achieving an accuracy only 2% off the maximum performance reported in the TAC 2010 challenge. The strong performance of the Cucerzan system on this data is surprising, given the results on the development data. On the TAC 2009 data, the Varma et al. system outperforms the Cucerzan system by 2% (see Table 14). There are a number of differences between the two data sets (as detailed in Table 3).

Table 22
Overall accuracy by genre and entity type (TAC 2010 test).

System	News			Web		
	ORG	GPE	PER	ORG	GPE	PER
NIL baseline	72.6	21.0	91.0	33.2	56.6	33.1
Title baseline	72.8	51.2	91.0	49.6	75.1	72.1
+Redirect baseline	74.8	65.6	97.0	80.4	**76.7**	82.9
Bunescu and Paşca (CosDAB)	77.6	65.6	97.2	87.6	65.5	86.9
Cucerzan (CosDAB)	**80.8**	68.4	**98.2**	86.4	60.2	87.6
Bunescu and Paşca	77.0	64.4	97.2	88.4	72.3	**89.6**
Cucerzan	77.2	**83.0**	**98.2**	83.6	71.9	88.0
Varma et al.	78.4	68.2	97.4	**90.0**	68.7	87.3

The 2009 data has more queries per entity, is skewed towards ORG queries and contains no web text. The 2010 test data is more varied and balanced, containing more entities overall (evenly balanced between KB and NIL) and an even distribution of queries by entity type. Acronyms comprise 15% of 2010 test queries versus 21% of 2009 queries and this may account for some performance loss for the Varma et al. [48] linker, which has specialised acronym processing.

10.1. Performance by genre and entity type

Table 22 contains *accuracy* scores broken down by genre (news or web) and entity type (ORG, GPE or PER). Rows correspond to the same systems reported in Table 21 above. The best scores in each column are in bold. The first thing to note is that no approach is consistently best across genres and entity types. This suggests that system combination by voting or entity-specific models may be worth investigating. Next, the percentage of NIL queries (as reflected in the NIL baseline scores) varies hugely across genre and entity types. In particular, the NIL percentage in web text is much lower than in news text for ORG and PER entities, but much higher for GPE entities.

There are two striking results about the behaviour of the Title + Redirect baseline system. First, the system performs near perfectly on PER entities in news text (97.0%). In part, this is probably attributable to the editorial standards associated with news, which results in PER entities mentioned in news generally being referred to using canonical forms. However, since the queries for the evaluation data set are not randomly sampled, it is not possible to quantify this observation. The second striking result is the fact that the Title + Redirect baseline outperforms all implemented systems on GPE entities in web text. This suggests that candidate generation is very noisy for these entities, which results in an especially difficult disambiguation problem. For ORG entities, systems with cosine disambiguators (including Varma et al.) are best in both news and web text. It is also interesting to note that there is very little variation in scores for PER entities, especially in news text.

Overall, our Cucerzan implementation is best for newswire, but does worse on web text. This holds for the cosine disambiguators as well as for the disambiguators from the literature. This suggests that the Cucerzan search strategy is tuned for more formal text. This may be attributed in part to the searcher's reliance on coreference and acronym handling, which are more accurate on text that follows the journalistic conventions for introducing new entities into discourse fairly unambiguously. For the Cucerzan disambiguator, the poorer performance of named entity recognition on web text is also likely to have the effect of introducing more noise into the document-level vector representations.

11. Discussion

Wikipedia is a rich source of data for natural language processing. Recently, it has been exploited for a number of information extraction tasks ranging from named entity recognition to relation extraction. This article explored the problem of entity linking, which disambiguates entity mentions by linking them to their Wikipedia page. This exciting new task moves beyond conventional named entity recognition where the output is a list of unnormalised entity mention strings. It shifts information extraction towards actionable semantic interpretation where objects in text are grounded to a node in an underlying knowledge base. The task opens up a range of applications from aggregation of information about a given entity across diverse structured, semi-structured and unstructured knowledge sources, to automated reasoning over extracted information.

The named entity linking task was first explored by Bunescu and Paşca [8] and Cucerzan [9] and has since been the focus of three shared tasks organised by the US National Institute of Standards and Technology as part of the Text Analysis Conferences (TAC) in 2009, 2010, and 2011. Previous approaches have largely focused on devising elaborate approaches to ranking a set of candidates, with the goal of promoting the true candidate to the top of the list. These assume a search strategy for generating a list of candidate entities, but previous work has not investigated candidate generation in detail. A notable exception is the top-scoring entry to the TAC 2009 shared task, which includes a highly tuned candidate generation strategy, but relies on a simple cosine similarity between the query context and the candidate Wikipedia page for ranking. This suggests that it is worthwhile to consider candidate generation strategies carefully.

A key theme across our results is that *baseline systems are difficult to beat*. Specifically, exact match lookup against page and redirect titles results in *accuracy* scores of 76.3% on the TAC 2009 test data and 79.4% on the TAC 2010 test data. This is due to the highly curated nature of Wikipedia, where commonly searched variations of names are very likely to have redirect or disambiguation pages. On the other hand, Wikipedia is a dynamic resource and redirect and disambiguation pages are thus likely to reflect changes in popularity of search terms over time. This has important implications for evaluation—*the version of Wikipedia used might have a strong effect on system performance, especially the recall of candidate generation*.

Another theme across our results is that *search strategies are extremely important*. Analysis of alias sources shows that page titles and redirect titles have very high precision; thus the Title + Redirect baseline is able to correctly return more than 50% of links in the TAC 2009 and 2010 data sets, while maintaining an NIL recall near 95%. Additionally, comparison across our searcher implementations highlighted the importance of coreference and acronym handling. Subtractive analysis of these components showed that they can lead to small improvements in *candidate recall* (+1.8 for Varma et al.). More importantly, they lead to an increase of approximately 5.5% in the percentage of candidate sets that include the correct answer (*candidate precision*) with a simultaneous decrease of approximately 0.8 in ambiguity as measured by the average candidate set size.

Detailed evaluation measures for candidate generation have proved useful for predicting subsequent performance on the end-to-end linking task. A searcher's *candidate recall*, for example, sets an upper bound on disambiguator performance. That is, the maximum KB *accuracy* obtainable by a disambiguator is equal to the *candidate recall* of the searcher proceeding it. Recent work reports dramatically higher candidate recall of 96.9% [28]. This is very promising and led to improved linker accuracy and warrants further investigation to determine the relative effect of its novel components: using token-based rather than exact match search, coreference filtering based on character overlap, and use of Google search. However, comparison of our search and linking results suggests that *improvements in* candidate recall *cannot come at the cost of* candidate precision*, and that search ambiguity needs to be carefully managed as well.* This is also supported by personal communication with Varma et al., in which they reported that, upon more detailed analysis, they found that the metaphone search employed in their TAC 2009 system actually reduced the final accuracy of their linker.

Our results highlight some interesting similarities and differences between the named entity and word sense disambiguation tasks. Both tasks have strong baselines related to *first sense heuristics*: one referent of a word or entity is much more common than the other possibilities, even when the number of other candidates is quite large. Wikipedia editors have adapted to this phenomenon by tuning article titles and redirects to capture the most likely intended meanings of common queries, which may be why the Title + Redirect baseline we present is so competitive. In both disambiguation tasks, the document contents are important clues for disambiguation, and simple methods based on bag-of-words models are fairly competitive. Early work on linking to Wikipedia [32] disambiguated arbitrary terminology, relating the task to word sense disambiguation. However, there is an important difference between named entity linking and conventional word sense disambiguation with WordNet: *the candidate senses for word sense disambiguation are provided directly, but candidate generation is critical for successful named entity linking.* The importance of this aspect of the problem has until now not been properly appreciated.

11.1. Recent literature

The implementation work we present is the start of a larger effort to perform a detailed comparison of various entity linking approaches within the same framework. A key development in the recent literature is the use learning-to-rank approaches. In addition to the Bunescu and Paşca [8] approach explored here, Dredze et al. [12] and Zheng et al. [54] use SVM^{rank} and ListNet respectively to incorporate a variety of features. Zheng et al. report 84.9% overall accuracy on the TAC 2010 test data. Another key development is the use of instance selection to generate training data from Wikipedia [53]. Zhang et al. [52] leverage this in achieving the current state-of-the-art performance of 86.1% on the TAC 2010 data.

Wikipedia structure has continued to drive new approaches, including those that eschew supervised machine learning. Han et al. [22] propose a generative probabilistic model based on entity, mention, and context statistics, which performs at 86% accuracy over the TAC 2009 data. Gottipati and Jiang [18] use language model-based information retrieval with NEmention and candidate context. This is particularly competitive on the variant of the TAC task in which Wikipedia text is not allowed. It obtains 85.2%, well above the top-ranking score of 77.9% from the official TAC 2010 results.

Wikipedia's link structure, in particular, has driven new approaches incorporating graph-based methods for NEL. This is the motivation behind citation overlap measures between candidates and unambiguous context entities [33,28,43,44]. More recent systems build a graph where vertices correspond to mentions and/or their entities and edges correspond to candidate entities for given mentions and/or entity–entity links from Wikipedia. Intuitively, highly connected regions represent the "topic" of a document and correct candidates should lie within these regions. Ploch [41] demonstrates that PageRank [7] values for candidate entities are a useful feature in their supervised ranking and NIL detection systems, leading to overall accuracy of 84.2% on the TAC 2009 data. Hachey et al. [20] show that degree centrality is better than PageRank, leading to performance of 85.5% on the TAC 2010 test data. And Guo et al. [19] show that degree centrality is better than a baseline similar to the cosine (CosDAB) baselines reported here, leading to performance of 82.4% on the TAC 2010 test data. Recent experiments on other data sets have also explored evidence propagation [22] and community detection [24].

12. Conclusion

Entity linking allows applications to compute with direct references to people, places and organisations, rather than potentially ambiguous or redundant character strings. As with other world knowledge problems, one important question about the task is what information a system must have access to in order to achieve satisfactory accuracy. This question is very difficult to answer by building a single system. Instead, a range of approaches must be evaluated in a single framework, with the ability to plug together different components and analyse them in detail.

We have presented the first systematic investigation of the entity linking problem, by implementing three of the canonical systems in the literature. We have performed the first direct comparison of these systems, analysed their errors in detail, and come to some surprising conclusions about the nature of the entity linking task.

We have found it useful to divide the entity linking task into two phases: search, and disambiguation. During the search phase the system proposes a set of candidates for a named entity mention to be linked to, which are then ranked by the disambiguator.

To our surprise, we found that much of the variation between the systems we considered was explained by the performance of their searchers. This was surprising because the literature on named entity linking has focused almost exclusively on disambiguation. The disambiguation task is arguably conceptually more interesting, since it lends itself to algorithmic solutions, and is related to the long-studied problem of word sense disambiguation. However, we have found that a simple vector space model performed surprisingly well compared to the more interesting disambiguation strategies we implemented.

Until now, it has been impossible to compare search and disambiguation strategies for entity linking directly, since only final accuracy figures have been available. Task accuracy is less informative, because it is unclear how ambitiously the searcher is proposing candidates for the disambiguator to rank. A conservative system with no disambiguation can perform surprisingly well, without offering any way to improve accuracy on the task in future. We have shown that state-of-the-art entity linking systems are pushing past this local maximum, but our results suggest that there is a long way to go on the difficult problem of determining which of a given set of candidates is the most likely referent of a named entity mention.

References

[1] J. Artiles, J. Gonzalo, S. Sekine, The SemEval 2007 WePS evaluation: Establishing a benchmark for the Web People Search task, in: Proceedings of the 4th International Workshop on Semantic Evaluations, 2007, pp. 64–69.
[2] J. Artiles, J. Gonzalo, S. Sekine, WePS 2 evaluation campaign: Overview of the Web People Search clustering task, in: Proceedings of the WWW Web People Search Evaluation Workshop, 2009.
[3] A. Bagga, B. Baldwin, Algorithms for scoring coreference chains, in: Proceedings of the LREC Linguistic Coreference Workshop, 1998, pp. 560–567.
[4] A. Bagga, B. Baldwin, Entity-based cross-document coreferencing using the vector space model, in: Proceedings of the 17th International Conference on Computational Linguistics, 1998, pp. 79–85.
[5] L. Bentivogli, P. Forner, C. Giuliano, A. Marchetti, E. Pianta, K. Tymoshenko, Extending English ACE 2005 corpus annotation with ground-truth links to Wikipedia, in: Proceedings of the Coling Workshop on the People's Web Meets NLP: Collaboratively Constructed Semantic Resources, 2010, pp. 19–27.
[6] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, S. Hellmann, DBpedia—a crystallization point for the web of data, Journal of Web Semantics 7 (2009) 154–165.
[7] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, in: Proceedings of the 7th International World Wide Web Conference, 1998, pp. 107–117.
[8] R. Bunescu, M. Paşca, Using encyclopedic knowledge for named entity disambiguation, in: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, 2006, pp. 9–16.
[9] S. Cucerzan, Large-scale named entity disambiguation based on Wikipedia data, in: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 2007, pp. 708–716.
[10] J.R. Curran, S. Clark, J. Bos, Linguistically motivated large-scale NLP with C&C and Boxer, in: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (demo), 2007, pp. 33–36.
[11] D. Day, J. Hitzeman, M. Wick, K. Crouch, M. Poesio, A corpus for cross-document co-reference, in: Proceedings of the 6th International Conference on Language Resources and Evaluation, 2008, pp. 23–31.
[12] M. Dredze, P. McNamee, D. Rao, A. Gerber, T. Finin, Entity disambiguation for knowledge base population, in: Proceedings of the 23rd International Conference on Computational Linguistics, 2010, pp. 277–285.
[13] A. Fader, S. Soderland, O. Etzioni, Scaling Wikipedia-based named entity disambiguation to arbitrary web text, in: Proceedings of the IJCAI Workshop on User-Contributed Knowledge and Artificial Intelligence, 2009, pp. 21–26.
[14] I.P. Fellegi, A.B. Sunter, A theory for record linkage, Journal of the American Statistical Association 64 (1969) 1183–1210.
[15] P. Ferragina, U. Scaiella, TAGME: on-the-fly annotation of short text fragments (by Wikipedia entities), in: Proceedings of the 19th International Conference on Information and Knowledge Management, 2010, pp. 1625–1628.
[16] W. Gale, K. Church, D. Yarowsky, Work on statistical methods for word sense disambiguation, in: Proceedings of the AAAI Fall Symposium on Intelligent Probabilistic Approaches to Natural Language, 1992, pp. 54–60.
[17] C.H. Gooi, J. Allan, Cross-document coreference on a large-scale corpus, in: Proceedings of the 7th Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2004, pp. 9–16.
[18] S. Gottipati, J. Jiang, Linking entities to a knowledge base with query expansion, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2011, pp. 804–813.
[19] Y. Guo, W. Che, T. Liu, S. Li, A graph-based method for entity linking, in: Proceedings of 5th International Joint Conference on Natural Language Processing, 2011, pp. 1010–1018.
[20] B. Hachey, W. Radford, J.R. Curran, Graph-based named entity linking with Wikipedia, in: Proceedings of the 12th International Conference on Web Information System Engineering, 2011, pp. 213–226.
[21] X. Han, L. Sun, A generative entity-mention model for linking entities with knowledge base, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, 2011, pp. 945–954.

[22] X. Han, L. Sun, J. Zhao, Collective entity linking in web text: A graph-based method, in: Proceedings of the 34th International Conference on Research and Development in Information Retrieval, 2011, pp. 765–774.
[23] L. Hirschman, M. Colosimo, A. Morgan, A. Yeh, Overview of BioCreAtIvE task 1B: Normalized gene lists, BMC Bioinformatics 6 (2005) S11.
[24] J. Hoffart, M.A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, G. Weikum, Robust disambiguation of named entities in text, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2011, pp. 782–792.
[25] W.C. Huang, S. Geva, A. Trotman, Overview of the INEX 2009 link the wiki track, in: Lecture Notes in Computer Science, vol. 6203, Springer-Verlag, Berlin/Heidelberg, 2010, pp. 312–323.
[26] T. Joachims, Training linear SVMs in linear time, in: Proceedings of the 12th International Conference on Knowledge Discovery and Data Mining, 2006, pp. 217–226.
[27] S. Kulkarni, A. Singh, G. Ramakrishnan, S. Chakrabarti, Collective annotation of Wikipedia entities in web text, in: Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining, 2009, pp. 457–466.
[28] J. Lehmann, S. Monahan, L. Nezda, A. Jung, Y. Shi, LCC approaches to knowledge base population at TAC 2010, in: Proceedings of the Text Analysis Conference, 2010.
[29] G.S. Mann, D. Yarowsky, Unsupervised personal name disambiguation, in: Proceedings of the 7th Conference on Computational Natural Language Learning, 2003, pp. 33–40.
[30] A. McCallum, K. Nigam, L.H. Ungar, Efficient clustering of high-dimensional data sets with application to reference matching, in: Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining, 2000, pp. 169–178.
[31] P. McNamee, H.T. Dang, H. Simpson, P. Schone, S.M. Strassel, An evaluation of technologies for knowledge base population, in: Proceedings of the 7th International Conference on Language Resources and Evaluation, 2010, pp. 369–372.
[32] R. Mihalcea, A. Csomai, Wikify!: Linking documents to encyclopedic knowledge, in: Proceedings of the 16th Conference on Information and Knowledge Management, 2007, pp. 233–242.
[33] D. Milne, I.H. Witten, Learning to link with Wikipedia, in: Proceedings of the 17th Conference on Information and Knowledge Management, 2008, pp. 509–518.
[34] M. Milosavljevic, J.Y. Delort, B. Hachey, B. Arunasalam, W. Radford, J.R. Curran, Automating financial surveillance, in: User Centric Media, in: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 40, Springer, Berlin/Heidelberg, 2010, pp. 305–311.
[35] A.A. Morgan, Z. Lu, X. Wang, A.M. Cohen, J. Fluck, P. Ruch, A. Divoli, K. Fundel, R. Leaman, J. Hakenberg, C. Sun, H. Liu, R. Torres, M. Krauthammer, W.W. Lau, H. Liu, C. Hsu, M. Schuemie, K.B. Cohen, L. Hirschman, Overview of BioCreative II gene normalization, Genome Biology 9 (2008) S3.
[36] R. Navigli, Word sense disambiguation: A survey, ACM Computing Surveys 41 (2009) 10:1–10:69.
[37] R. Navigli, S.P. Ponzetto, Babelnet: Building a very large multilingual semantic network, in: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 2010, pp. 216–225.
[38] NIST, The ACE 2005 (ACE05) evaluation plan, http://www.itl.nist.gov/iad/mig/tests/ace/2005/doc/ace05-evalplan.v3.pdf, 2005.
[39] C. Niu, W. Li, R.K. Srihari, Weakly supervised learning for cross-document person name disambiguation supported by information extraction, in: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 2004, pp. 597–604.
[40] J. Nothman, T. Murphy, J.R. Curran, Analysing Wikipedia and gold-standard corpora for NER training, in: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, 2009, pp. 612–620.
[41] D. Ploch, Exploring entity relations for named entity disambiguation, in: Proceedings of the ACL Student Session, 2011, pp. 18–23.
[42] S.P. Ponzetto, M. Strube, Taxonomy induction based on a collaboratively built knowledge repository, Artificial Intelligence 175 (2011) 1737–1756.
[43] W. Radford, B. Hachey, J. Nothman, M. Honnibal, J.R. Curran, CMCRC at TAC10: Document-level entity linking with graph-based reranking, in: Proceedings of the Text Analysis Conference, 2010.
[44] L. Ratinov, D. Roth, D. Downey, M. Anderson, Local and global algorithms for disambiguation to Wikipedia, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, 2011, pp. 1375–1384.
[45] C. Sauper, R. Barzilay, Automatically generating Wikipedia articles: A structure-aware approach, in: Proceedings of the Joint 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing, 2009, pp. 208–216.
[46] F.M. Suchanek, G. Kasneci, G. Weikum, Yago: A large ontology from Wikipedia and WordNet, Journal of Web Semantics 6 (2008) 203–217.
[47] E.F. Tjong Kim Sang, F. De Meulder, Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition, in: Proceedings of the 7th Conference on Natural Language Learning, 2003, pp. 142–147.
[48] V. Varma, P. Bysani, K. Reddy, V. Bharat, Santosh G.S.K., K. Kumar, S. Kovelamudi, Kiran Kumar N., N. Maganti, IIIT Hyderabad at TAC 2009, in: Proceedings of the Text Analysis Conference, 2009.
[49] W.E. Winkler, Overview of record linkage and current research directions, Technical Report, Bureau of the Census, 2006.
[50] K. Woodsend, M. Lapata, Learning to simplify sentences with quasi-synchronous grammar and integer programming, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2011, pp. 409–420.
[51] F. Wu, D.S. Weld, Autonomously semantifying Wikipedia, in: Proceedings of the 16th Conference on Information and Knowledge Management, 2007, pp. 41–50.
[52] W. Zhang, C.S. Sim, J. Su, C.L. Tan, Entity linking with effective acronym expansion, instance selection and topic modelling, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, 2011, pp. 1909–1914.
[53] W. Zhang, J. Su, C.L. Tan, W.T. Wang, Entity linking leveraging automatically generated annotation, in: Proceedings of the 23rd International Conference on Computational Linguistics, 2010, pp. 1290–1298.
[54] Z. Zheng, F. Li, M. Huang, X. Zhu, Learning to link entities with knowledge base, in: Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010, pp. 483–491.