

Entity Linking: Finding Extracted Entities in a Knowledge Base

Delip Rao¹, Paul McNamee², and Mark Dredze^{1,2}

¹ Department of Computer Science
Johns Hopkins University
(delip|mdredze)@cs.jhu.edu

² Human Language Technology Center of Excellence
Johns Hopkins University
paul.mcnamee@jhuapl.edu

Abstract. In the menagerie of tasks for information extraction, entity linking is a new beast that has drawn a lot of attention from NLP practitioners and researchers recently. Entity Linking, also referred to as record linkage or entity resolution, involves aligning a textual mention of a named-entity to an appropriate entry in a knowledge base, which may or may not contain the entity. This has manifold applications ranging from linking patient health records to maintaining personal credit files, prevention of identity crimes, and supporting law enforcement. We discuss the key challenges present in this task and we present a high-performing system that links entities using max-margin ranking. We also summarize recent work in this area and describe several open research problems.

Keywords: Entity Linking, Record Linkage, Entity Resolution, Knowledge Base Population, Entity Disambiguation, Named Entities

1 Introduction

Information extraction involves the processing of natural language text to produce structured knowledge, suitable for storage in a database for later retrieval or automated reasoning. An active area of research for over twenty years, the community has developed several core information extraction tasks that comprise an extraction pipeline.

Named Entity Recognition: Identify boundaries of named entities in text and classify the tokens into a predefined set of named entities, such as people, organizations and locations. See for example [40, 2, 28, 9, 11], and the review article by Nadeau and Sekine [32].

Coreference Resolution: Group two or more named entities and other anaphoras in a document or a set of documents that refer to the same real world entity. For example, “Bush”, “Mr. President”, “G. W. Bush”, and “George Bush” occurring in a set of documents might refer to the same entity [3, 12, 46, 33].

Relation Extraction: Given two named entities, identify relationships between the entities expressed in the text. For instance, given two person names in news documents about crime and violence, identify the victim and the perpetrator [4, 44]. Most relation extraction methods can be classified as open- or closed-domain depending on the restrictions on extractable relations. Closed domain systems extract a fixed set of relations while in open-domain systems, the number and type of relations are unbounded.

This document-centric view of information extraction has received considerable attention. However, the end result, a group of entities and relations, often are not the only structured knowledge product. In a development environment, new extractions must be merged with previously extracted information, often stored in a structured information database, a knowledge base (KB). This last step is critical for automatic knowledge base population, which requires linking mentions in text to entries in a KB, determining information duplication between the text and KB, exploiting existing knowledge in improving information extraction, and detecting when to create new entries in the knowledge base. These challenges are exacerbated by the scale of the data, often involving hundreds of thousands of documents and several million entities.

To the discerning human eye, the “Bush” in “Mr. Bush left for the Zurich environment summit in Air Force One.” is clearly the US president. Further context may reveal him to be the 43rd president, George W. Bush, and not the 41st president, George H. W. Bush. The ability to disambiguate a polysemous entity mention or infer that two orthographically different mentions are the same entity is crucial in updating an entity’s KB record. This task has been variously called entity disambiguation, record linkage, or entity linking. When performed without a KB, entity disambiguation reduces to the traditional document coreference resolution problem where entity mentions either within the same document or across multiple documents are clustered together, where each cluster corresponds to a single real world entity. The emergence of large scale publicly available KBs like Wikipedia and DBpedia has spurred an interest in linking textual entity references to their entries in these public KBs. Bunescu and Pasca [7] and Cucerzan [10] presented important pioneering work in this area, but suffer from several limitations including Wikipedia specific dependencies, scale, and the assumption of a KB entry for each entity.

In this chapter, we review some common approaches to entity disambiguation and discuss in detail an entity disambiguation system for linking entity mentions (also called an entity linking query) to an entry in a knowledge base or declare if no such entry exists. We adopt a supervised machine learning approach, where each of the possible entities contained in the KB are scored for a possible match to the query entity. Our system is designed for open domains, where a large percentage of entities will not be linkable since they do not appear in the knowledge base. For this scenario, our system learns when to withhold a link when an entity has no matching KB entry, a task that has largely been neglected in prior research in cross-document entity coreference. We also describe techniques to deal with large knowledge bases, such as Wikipedia and DBpedia, which contain

millions of entries. Our system produces high quality predictions compared with recent work on this task.

2 Prior Art

Information extraction is concerned with both identifying structured information in text and disambiguating extracted information and entities. The ambiguity of entity names, especially on large corpora like the Web or citations in scholarly articles, has served to motivate research on entity resolution. To address ambiguity in personal name search, Mann and Yarowsky [26] disambiguates person names using biographic facts, like birth year, occupation and affiliation. When present in text, biographic facts extracted using regular expressions help disambiguation. More recently, the Web People Search Task, see [1] for example, clustered web pages for entity disambiguation.

The related task of cross document coreference resolution has been addressed by several researchers starting from Bagga and Baldwin [3]. Poesio et al [35] built a cross document coreference system using features from encyclopedic sources like Wikipedia. This continues to be a popular task [36] that considers new data sets [20]. Entity linking has been scaled to consider hundreds of thousands of unique entities, whereas operating on this scale is a challenge for cross document coreference resolution. Recent approaches to scaling this task have included distributed graphical models over a compute cluster [42] and a streaming coreference algorithm [38]. Successful coreference resolution is insufficient for correct entity linking, as the coreference chain must still be correctly mapped to the proper KB entry.

A related task is within document coreference, or anaphora resolution, in which co-referent named entity, pronominal, and nominal mentions are linked together in an entity chain. This task has a long history in the NLP community [34] which still receives significant attention [37, 15, 43]. Interestingly, coreference systems have now been used as part of larger information extraction systems, such as relation extraction [17].

By comparison, entity linking is a recent task. The earliest work on the task by Bunescu and Pasca [7] and Cucerzan [10] aims to link entity mentions to their corresponding topic pages in Wikipedia. These authors do not use the term entity linking and they take different approaches. Cucerzan uses heuristic rules and Wikipedia disambiguation markup to derive mappings from surface forms of entities to their Wikipedia entries. For each entity in Wikipedia, a context vector is derived as a prototype for the entity and these vectors are compared (via dot-product) with the context vectors of unknown entity mentions. His work assumes that all entities have a corresponding Wikipedia entry, but this assumption fails for a significant number of entities in news articles and even more for other genres, like blogs. Bunescu and Pasca on the other hand suggest a simple method to handle entities not in Wikipedia by learning a threshold to decide if the entity is not in Wikipedia. Both works mentioned rely on Wikipedia-specific annotations, such as category hierarchies and disambiguation links.

The Entity Linking problem not only disambiguates entity mentions that occur in text but also link these mentions to entries in the knowledge base. This is the focus of this chapter. Since the Text Analytics Conference on Knowledge Base Population (TAC-KBP) included the task of entity linking [30], the task has grown in popularity with many different approaches [21, 49]. Examples include the use of information retrieval techniques for retrieving the correct KB entry, such as query expansion [18], and generative clustering models for entities in text based on KB entries [19].

The work described in this paper was developed as one of the first entity linking systems. Subsequent systems have built on our approach [24, 48, 19, 47].

3 Entity Linking

We now describe the details of building such a system and summarize other systems built for this task. We define *entity linking* as matching a textual entity mention, possibly identified by a named entity recognizer, to a KB entry, such as a Wikipedia page that is a canonical entry for that entity. An entity linking *query* is a request to link a textual entity mention in a given document to an entry in a KB. The system can either return a matching entry or NIL to indicate there is no matching entry. In this work we focus on linking organizations, geo-political entities and persons to a Wikipedia derived KB. While the problem is applicable for any language, in this paper we restrict our attention to matching English names to an English knowledge base.

3.1 Key Issues

There are 3 challenges to entity linking:

Name Variations. An entity often has multiple mention forms, including abbreviations (Boston Symphony Orchestra vs. BSO), shortened forms (Osama Bin Laden vs. Bin Laden), alternate spellings (Osama vs. Ussamah vs. Oussama), and aliases (Osama Bin Laden vs. Sheikh Al-Mujahid). Entity linking must find an entry despite changes in the mention string.

Entity Ambiguity. A single mention, like Springfield, can match multiple KB entries, as many entity names, like people and organizations, tend to be polysemous.

Absence. Processing large text collections virtually guarantees that many entities will not appear in the KB (NIL), even for large KBs.

The combination of these challenges makes entity linking especially challenging. Consider an example of “William Clinton.” Most readers will immediately think of the 42nd US president. However, as of this writing, the only two William Clintons in Wikipedia are “William de Clinton” the 1st Earl of Huntingdon, and “William Henry Clinton” the British general. The page for the 42nd US president is actually “Bill Clinton”. An entity linking system must decide if either

of the William Clintons are correct, even though neither are exact matches. If the system determines neither matches, should it return NIL or the variant “Bill Clinton”? If variants are acceptable, then perhaps “Clinton, Iowa” or “DeWitt Clinton” should be acceptable answers?

3.2 Contributions

We address these entity linking challenges.

Robust Candidate Selection. Our system is flexible enough to find name variants but sufficiently restrictive to produce a manageable candidate list despite a large-scale KB.

Ranking and Features for Entity Disambiguation. We developed a rich and extensible set of features based on the entity mention, the source document, and the KB entry. We use a machine learning ranker to score each candidate.

Learning NILs. We modify the ranker to learn NIL predictions, which obviates hand tuning and importantly, admits use of additional features that are indicative of NIL.

Our contributions differ from previous efforts [7, 10] in several important ways. First, previous efforts depend on Wikipedia markup for significant performance gains. We make no such assumptions, although we show that optional Wikipedia features lead to a slight improvement. Second, Cucerzan does not handle NILs while Bunescu and Pasca address them by learning a threshold. Our approach *learns* to predict NIL in a more general and direct way. Third, we develop a rich feature set for entity linking that can work with any KB. Finally, we apply a novel finite state machine method for *learning* name variations.³

The remaining sections describe the candidate selection stage, our ranking algorithm and features, and our novel approach to learning NILs.

4 Candidate Selection for Name Variants

The first system component addresses the challenge of name variants. As the KB contains a large number of entries (818,000 entities, of which 35% are PER, ORG or GPE), we require an efficient selection of the relevant candidates for a query.

Previous approaches used Wikipedia markup for filtering – only using the top-k page categories [7] – which is limited to Wikipedia and does not work for general KBs. We first consider a KB independent approach to selection that also allows for tuning candidate set size. This involves a linear pass over KB entry names (Wikipedia page titles): a naive implementation took two minutes per query.

³ <http://www.clsp.jhu.edu/markus/fstrain>

4.1 Brute Force Candidate Selection

For a given query, the system selects KB entries using the following approach:

- Titles that are exact matches for the mention.
- Titles that are wholly contained in or contain the mention (e.g., *Nationwide* and *Nationwide Insurance*).
- The first letters of the entity mention match the KB entry title (e.g., *OA* and *Olympic Airlines*).
- The title matches a known alias for the entity (aliases described in Section 5.2).
- The title has a strong string similarity score with the entity mention. We include several measures of string similarity, including: character Dice score > 0.9 , skip bigram Dice score > 0.6 , and Hamming distance ≤ 2 .

We did not optimize the thresholds for string similarity, but these could obviously be tuned to minimize the candidate sets and maximize recall. For a comprehensive survey on string similarity metrics for duplicate names, we refer the reader to [14].

All of the above features are general for any KB. However, since our evaluation used a KB derived from Wikipedia, we included a few Wikipedia specific features. We added an entry if its Wikipedia page appeared in the top 20 Google results for a query.

On the training dataset (Section 7) the selection system attained a recall of 98.8% and produced candidate lists that were three to four orders of magnitude smaller than the KB. Some recall errors were due to inexact acronyms: ABC (Arab Banking; ‘Corporation’ is missing), ASG (Abu Sayyaf; ‘Group’ is missing), and PCF (French Communist Party; French reverses the order of the pre-nominal adjectives). We also missed International Police (Interpol) and Becks (David Beckham; Mr. Beckham and his wife are collectively referred to as ‘Posh and Becks’).

4.2 Sublinear Candidate Selection

Our previously described candidate selection relied on a linear pass over the KB, but we seek more efficient methods. We observed that many of the above string similarity filters, such as aliases and exact string matches, can be pre-computed and stored in an index, resulting in significant speedups. Additionally, the skip bigram Dice score can be computed using an index of skip bigrams to KB titles, removing from consideration the vast of titles which have no skip bigram overlap with the query. Other string similarity scores were omitted without significantly hurting the recall of the filtering stage. These changes collectively enable us to avoid a linear pass over the KB. Finally we obtained speedups by serving the KB concurrently using 4 processes, each of which execute queries against a portion of the KB. This allows parallelization can be extended for larger KBs. The results from each process are collected for the second ranking stage. We implemented this approach in Python and our system achieved up to an $80\times$ speedup

compared to naive implementation, serving each query in under two *seconds* on average. Recall was nearly identical to the full system described above: only two more queries failed. Additionally, more than 95% of the processing time was consumed by Dice score computation, which was only required to correctly retrieve less than 4% of the training queries. Omitting the Dice computation yielded results in a few milliseconds on average. A related approach is that of canopies for scaling clustering for large amounts of bibliographic citations [29]. In contrast, our setting focuses on alignment vs. clustering mentions, for which overlapping partitioning approaches like canopies are applicable.

5 Entity Linking as Ranking

We consider a supervised machine learning approach to entity linking. Given a query represented by a D dimensional vector \mathbf{x} , where $\mathbf{x} \in \mathbb{R}^D$, and we aim to select a single KB entry y , where $y \in \mathcal{Y}$, a set of possible KB entries for this query produced by the selection system above, which ensures that \mathcal{Y} is small. The i th query is given by the pair $\{\mathbf{x}_i, y_i\}$, where we assume at most one correct KB entry. Using these training examples, we can learn a system that produces the correct y for each query.

To evaluate each candidate KB entry in \mathcal{Y} we create feature functions of the form $f(\mathbf{x}, y)$, dependent on both the example \mathbf{x} (document and entity mention) and the KB entry y . The features address name variants and entity disambiguation. We categorize the features as atomic features and combination features. Atomic features are derived directly from the named entity in question and its context while combination features are logical expressions of atomic features in conjunctive normal form (CNF).

One natural approach to learning would be classification, in which each possible $y \in \mathcal{Y}$ is classified as being either correct or incorrect. However, such an approach enforces strong constraints: we not only require the correct KB entry to be classified positively, but all other answers to be classified negatively. Additionally, we can expect very unbalanced training, in which the vast majority of possible answers are incorrect. Furthermore, it is unclear how to select a correct answer at test time when multiple KB entries can be classified as correct.

Instead, we select a single correct candidate for a query using a supervised machine learning ranker. A ranker will create an ordering over a set of answers \mathcal{Y} given a query. Typically, the resulting order over all items is important, such as ranking results for web search queries. In our setting, we assume only a single correct answer and therefore impose a looser requirement, that the correct answer be ranked highest. This formulation addresses several of the challenges of binary classification. We require only that relative scores be ordered correctly, not that each entry be given a label of correct/incorrect. Training is balanced as we have a single ranking example for each query. And finally, we simply select the highest ranked entry as correct, no matter its score.

We take a maximum margin approach to learning: the correct KB entry y should receive a higher score than all other possible KB entries $\hat{y} \in \mathcal{Y}, \hat{y} \neq y$

plus some margin γ . This learning constraint is equivalent to the ranking SVM algorithm of Joachims [22], where we define an ordered pair constraint for each of the incorrect KB entries \hat{y} and the correct entry y . Since we have a preference only for the relative ordering of a single entry compared to all others, we introduce a linear number of constraints for learning. Only the position of the correct entry is important. Training sets parameters such that $\text{score}(y) \geq \text{score}(\hat{y}) + \gamma$. We used the library SVM^{rank} to solve this optimization problem.⁴ We used a linear kernel, set the slack parameter C as 0.01 times the number of training examples, and take the loss function as the total number of swapped pairs summed over all training examples. While previous work used a custom kernel, we found a linear kernel just as effective with our features. This has the advantage of efficiency in both training and prediction⁵ – important considerations in a system meant to scale to millions of KB entries.

5.1 Features for Entity Disambiguation

200 atomic features represent \mathbf{x} based on each candidate query/KB pair. Since we used a linear kernel, we explicitly combined certain features (e.g., acronym-match AND known-alias) to model correlations. This included combining each feature with the predicted type of the entity, allowing the algorithm to learn prediction functions specific to each entity type. With feature combinations, the total number of features grew to 26,569. The next sections provide an overview; for a detailed list see [31].

5.2 Features for Name Variants

Variation in entity name has long been recognized as a bane for information extraction systems. Poor handling of entity name variants results in low recall. We describe several features ranging from simple string match to finite state transducer matching.

String Equality. If the query name and KB entry name are identical, this is a strong indication of a match, and in our KB entry names are distinct. However, similar or identical entry names that refer to distinct entities are often qualified with parenthetical expressions or short clauses. As an example, “London, Kentucky” is distinguished from “London, Ontario”, “London, Arkansas”, “London (novel)”, and “London”. Therefore, other string equality features were used, such as whether names are equivalent after some transformation. For example, “Baltimore” and “Baltimore City” are exact matches after removing a common GPE word like city; “University of Vermont” and “University of VT” match if VT is expanded.

⁴ www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

⁵ [7] report learning tens of thousands of support vectors with their “taxonomy” kernel while a linear kernel represents all support vectors with a single weight vector, enabling faster training and prediction.

Approximate String Matching. Many entity mentions will not match full names exactly. We added features for character Dice, skip bigram Dice, and left and right Hamming distance scores. Features were set based on quantized scores. These were useful for detecting minor spelling variations or mistakes. Features were also added if the query was wholly contained in the entry name, or vice-versa, which was useful for handling ellipsis (e.g., “United States Department of Agriculture” vs. “Department of Agriculture”). We also included the ratio of the recursive longest common subsequence [8] to the shorter of the mention or entry name, which is effective at handling some deletions or word reorderings (e.g., “Li Gong” and “Gong Li”). Finally, we checked whether all of the letters of the query are found in the same order in the entry name (e.g., “Univ Wisconsin” would match “University of Wisconsin”).

Acronyms. Features for acronyms, using dictionaries and partial character matches, enable matches between “MIT” and “Madras Institute of Technology” or “Ministry of Industry and Trade.”

Aliases. Many aliases or nicknames are non-trivial to guess. For example JAVA is the stock symbol for Sun Microsystems, and “Ginger Spice” is a stage name of Geri Halliwell. A reasonable way to do this is to employ a dictionary and alias lists that are commonly available for many domains⁶.

FST Name Matching. Another measure of surface similarity between a query and a candidate was computed by training finite-state transducers similar to those described in [13]. These transducers assign a score to any string pair by summing over all alignments and scoring all contained character n -grams; we used n -grams of length 3 and less. The scores are combined using a global log-linear model. Since different spellings of a name may vary considerably in length (e.g., *J Miller* vs. *Jennifer Miller*) we eliminated the limit on consecutive insertions used in previous applications.⁷

5.3 Wikipedia Features

Most of our features do not depend on Wikipedia markup, but it is reasonable to include features from KB properties. Our feature ablation study shows that dropping these features causes a small but statistically significant performance drop.

WikiGraph statistics. We added features derived from the Wikipedia graph structure for an entry, like indegree of a node, outdegree of a node, and Wikipedia page length in bytes. These statistics favor common entity mentions over rare ones.

⁶ We used multiple lists, including class-specific lists (i.e., for PER, ORG, and GPE) lists extracted from Freebase [5] and Wikipedia redirects. PER, ORG, and GPE are the commonly used terms for entity types for people, organizations and geo-political regions respectively.

⁷ Without such a limit, the objective function may diverge for certain parameters of the model; we detect such cases and learn to avoid them during training.

Wikitalogy. KB entries can be indexed with human or machine generated meta-data consisting of keywords or categories in a domain-appropriate taxonomy. Using a system called *Wikitalogy*, Syed et al [45] investigated use of ontology terms obtained from the explicit category system in Wikipedia as well as relationships induced from the hyperlink graph between related Wikipedia pages. Following this approach we computed top-ranked categories for the query documents and used this information as features. If none of the candidate KB entries had corresponding highly-ranked Wikitalogy pages, we used this as a NIL feature (Section 6).

5.4 Popularity

Although it may be an unsafe bias to give preference to common entities, we find it helpful to provide estimates of entity popularity to our ranker as others have done [16]. Apart from the graph-theoretic features derived from the Wikipedia graph, we also used Google’s PageRank by adding features indicating the rank of the KB entry’s corresponding Wikipedia page in a Google query for the target entity mention.

5.5 Document Features

The mention document and text associated with a KB entry contain context for resolving ambiguity.

Entity Mentions. Some features were based on presence of names in the text: whether the query appeared in the KB text and the entry name in the document. Additionally, we used a named-entity tagger and relation finder, SERIF [6], which identified name and nominal mentions that were deemed co-referent with the entity mention in the document, and tested whether these nouns were present in the KB text. Without the NE analysis, accuracy on non-NIL entities dropped 4.5%.

KB Facts. KB nodes contain infobox attributes (or facts); we tested whether the fact text was present in the query document, both locally to a mention, or anywhere in the text. Although these facts were derived from Wikipedia infoboxes, they could be obtained from other sources as well.

Document Similarity We measured similarity between the query document and the KB text in two ways: cosine similarity with TF/IDF weighting [39]; and using the Dice coefficient over bags of words. IDF values were approximated using counts from the Google 5-gram dataset as by [23].

Entity Types. Since the KB contained types for entries, we used these as features as well as the predicted NE type for the entity mention in the document text. Additionally, since only a small number of KB entries had PER, ORG, or GPE types, we also inferred types from Infobox class information to attain 87% coverage in the KB. This was helpful for discouraging selection of eponymous entries named after famous entities (e.g., the former U.S. president vs. “John F. Kennedy International Airport”).

5.6 Feature Combinations

To take into account feature dependencies we created combination features by taking the cross-product of a small set of diverse features. The attributes used as combination features included entity type; a popularity based on Google’s rankings; document comparison using TF/IDF; coverage of co-referential nouns in the KB node text; and name similarity. The combinations were cascaded to allow arbitrary feature conjunctions. Thus it is possible to end up with a feature *kbtype-is-ORG AND high-TFIDF-score AND low-name-similarity*. The combined features increased the number of features from roughly 200 to 26,000.

6 Predicting NIL Mentions

So far we have assumed that each example has a correct KB entry; however, when run over a large corpus, such as news articles, we expect a significant number of entities will not appear in the KB. Hence it will be useful to predict NILs.

We *learn* when to predict NIL using the SVM ranker by augmenting \mathcal{Y} to include NIL, which then has a single feature unique to NIL answers. It can be shown that (modulo slack variables) this is equivalent to learning a single threshold τ for NIL predictions as in [7].

Incorporating NIL into the ranker has several advantages. First, the ranker can set the threshold optimally without hand tuning. Second, since the SVM scores are relative within a single example and cannot be compared across examples, setting a single threshold is difficult. Third, a threshold sets a uniform standard across all examples, whereas in practice we may have reasons to favor a NIL prediction in a given example. We design features for NIL prediction that cannot be captured in a single parameter.

Integrating NIL prediction into learning means we can define arbitrary features indicative of NIL predictions in the feature vector corresponding to NIL. For example, if many candidates have good name matches, it is likely that one of them is correct. Conversely, if no candidate has high entry-text/article similarity, or overlap between facts and the article text, it is likely that the entity is absent from the KB. We included several features, such as a) the max, mean, and difference between max and mean for 7 atomic features for all KB candidates considered, b) whether any of the candidate entries have matching names (exact and fuzzy string matching), c) whether any KB entry was a top Wikitology match, and d) if the top Google match was not a candidate.

7 Evaluation

We evaluated our system on two datasets: the Text Analysis Conference (TAC) track on Knowledge Base Population (TAC-KBP) [30] and the newswire data used by Curcerzan in [10] (Microsoft News Data).

	Micro-Averaged				Macro-Averaged			
	<i>Best</i>	<i>Median</i>	<i>All Feats</i>	<i>Best Feats</i>	<i>Best</i>	<i>Median</i>	<i>All Feats</i>	<i>Best Feats</i>
All	0.8217	0.7108	0.7984	0.7941	0.7704	0.6861	0.7695	0.7704
non-NIL	0.7725	0.6352	0.7063	0.6639	0.6696	0.5335	0.6097	0.5593
NIL	0.8919	0.7891	0.8677	0.8919	0.8789	0.7446	0.8464	0.8721

Table 1. Micro and macro-averaged accuracy for TAC-KBP data compared to best and median reported performance. Results are shown for all features as well as removing a small number of features using feature selection on development data.

Since our approach relies on supervised learning, we begin by constructing our own training corpus.⁸ We highlighted 1496 named entity mentions in news documents (from the TAC-KBP document collection) and linked these to entries in a KB derived from Wikipedia infoboxes.⁹ We added to this collection 119 sample queries from the TAC-KBP data. The total of 1615 training examples included 539 (33.4%) PER, 618 (38.3%) ORG, and 458 (28.4%) GPE entity mentions. Of the training examples, 80.5% were found in the KB, matching 300 unique entities. This set has a higher number of NIL entities than did Bunescu & Pasca [7] (10%) but lower than the TAC-KBP test set (43%).

All system development was done using a train (908 examples) and development (707 examples) split. The TAC-KBP and Microsoft News data sets were held out for final tests. A model trained on all 1615 examples was used for experiments.

7.1 TAC-KBP 2009 Experiments

In 2009, 2010 and 2011, NIST conducted evaluations of entity linking technologies as part of the Text Analysis Conference (TAC). The Knowledge Base Population track (TAC-KBP) focused on two subtasks: linking mentions of entities to a standard KB, and gleaning novel attributes about and relationships between entities from a large corpus. In the entity linking subtask, each query consisted of a name string and a reference document that contained the name string and provided context to help determine which KB entity is being referred to. Each query was either a person (PER), organization (ORG), or geo-political entity (GPE; essentially an inhabited location) but the entity type was not provided in the query. A breakdown of queries by type is given in Table 4. In 2009 queries were not balanced by entity type or presence in the KB (*e.g.*, organizations accounted for a majority of the of the queries - 69%), but in 2010 a more uniform distribution was created. Persons and organizations were more likely to be absent than GPEs, which have broad coverage in the KB, as they do in Wikipedia. As of this work’s publication, the 2011 task is underway.

We evaluated our approach on the 2009 data and describe the data in detail. The KB is derived from English Wikipedia pages that contained an infobox.

⁸ Data available from www.dredze.com

⁹ <http://en.wikipedia.org/wiki/Help:Infobox>

Entries contain basic descriptions (article text) and attributes. The TAC-KBP query set contains 3904 entity mentions for 560 distinct entities; entity type was only provided for evaluation. The majority of queries were for organizations (69%). Most queries were missing from the KB (57%). 77% of the distinct GPEs in the queries were present in the KB, but for PERs and ORGs these percentages were significantly lower, 19% and 30% respectively.

Fictional entities, which are well-covered in Wikipedia, and time-sensitive entities (*e.g.*, ORGs with dynamic membership such as the US Olympic men’s ice hockey team) were deliberately excluded as targets. Also prohibited were names that can refer to a group of entities (*e.g.*, Blue Devils might refer to any of Duke University’s athletic teams). Care was taken to avoid using documents where the target entity name was internally ambiguous. Additional details about the target selection process are described in Simpson *et al.* [41]. In Table 5 several of the 2009 entity linking queries are presented, along with KB node that was judged to be correct.

Table 1 shows results on TAC-KBP data using all of our features as well a subset of features based on feature selection experiments on development data. We include scores for both micro-averaged accuracy – averaged over all queries – and macro-averaged accuracy – averaged over each unique entity – as well as the best and median reported results for these data [30]. We obtained the best reported results for macro-averaged accuracy, as well as the best results for NIL detection with micro-averaged accuracy, which shows the advantage of our approach to learning NIL. See [31] for additional experiments.

The candidate selection phase obtained a recall of 98.6%, similar to that of development data. Missed candidates included *Iron Lady*, which refers metaphorically to Yulia Tymoshenko, *PCC*, the Spanish-origin acronym for the Cuban Communist Party, and *Queen City*, a former nickname for the city of Seattle, Washington. The system returned a mean of 76 candidates per query, but the median was 15 and the maximum 2772 (*Texas*). In about 10% of cases there were four or fewer candidates and in 10% of cases there were more than 100 candidate KB nodes. We observed that ORGs were more difficult, due to the greater variation and complexity in their naming, and that they can be named after persons or locations.

7.2 Feature Effectiveness

We performed two feature analyses on the TAC-KBP data: an additive study – starting from a small baseline feature set used in candidate selection we add feature groups and measure performance changes (omitting feature combinations), and an ablative study – starting from all features, remove a feature group and measure performance.

Table 2 shows the most significant features in the feature addition experiments. The baseline includes only features based on string similarity or aliases and is not effective at finding correct entries and strongly favors NIL predictions. Inclusion of features based on analysis of named-entities, popularity measures

<i>Class</i>	<i>All</i>	<i>non-NIL</i>	<i>NIL</i>
Baseline	0.7264	0.4621	0.9251
Acronyms	0.7316	0.4860	0.9161
NE Analysis	0.7661	0.7181	0.8022
Google	0.7597	0.7421	0.7730
Doc/KB Text Similarity	0.7313	0.6699	0.7775
Wikitology	0.7318	0.4549	0.9399
All	0.7984	0.7063	0.8677

Table 2. Additive analysis: micro-averaged accuracy.

(e.g., Google rankings), and text comparisons provided the largest gains. Although the overall changes are fairly small the changes in non-NIL precision are much larger.

The ablation study showed considerable redundancy across feature groupings. In several cases, performance could have been slightly improved by removing features. Removing all feature combinations would have improved overall performance to 81.05% by gaining on non-NIL for a small decline on NIL detection.

7.3 Experiments on Microsoft News Data

We downloaded the evaluation data used in [10]¹⁰: 20 news stories from MSNBC with 642 entity mentions manually linked to Wikipedia and another 113 mentions not having any corresponding link to Wikipedia.¹¹ A significant percentage of queries were not of type PER, ORG, or GPE (e.g., “Christmas”). SERIF assigned entity types and we removed 297 queries not recognized as entities (counts in Table 3).

We learned a new model on the training data above using a reduced feature set to increase speed.¹² Using our fast candidate selection system, we resolved each query in 1.98 seconds (median). Query processing time was proportional to the number of candidates considered. We selected a median of 13 candidates for PER, 12 for ORG and 102 for GPE. Accuracy results are in Table 3. The high results reported for this dataset over TAC-KBP is primarily because we perform very well in predicting popular and rare entries – both of which are common in newswire text.

¹⁰ <http://research.microsoft.com/en-us/um/people/silviu/WebAssistant/TestData/>

¹¹ One of the MSNBC news articles is no longer available so we used 759 total entities.

¹² We removed Google, FST and conjunction features which reduced system accuracy but increased performance.

	Num. Queries		Accuracy		
	<i>Total</i>	<i>Nil</i>	<i>All</i>	<i>non-NIL</i>	<i>NIL</i>
NIL	452	187	0.4137	0.0	1.0
GPE	132	20	0.9696	1.00	0.8000
ORG	115	45	0.8348	0.7286	1.00
PER	205	122	0.9951	0.9880	1.00
All	452	187	0.9469	0.9245	0.9786
Cucerzan (2007)			0.914	-	-

Table 3. Micro-average results for Microsoft data.

<i>Type</i>	<i>2009 Queries</i>			<i>2010 Queries</i>		
	<i>Total</i>	<i>KB</i>	<i>Missing</i>	<i>Total</i>	<i>KB</i>	<i>Missing</i>
PER	627	255	372	751	213	538
ORG	2710	1013	1697	750	304	446
GPE	567	407	160	749	503	246
All	3904	1675	2229	2250	1020	1230

Table 4. Number of queries/entities by type and presence in the KB.

One issue with our KB was that it was derived from infoboxes in Wikipedia’s Oct 2008 version which has both new entities,¹³ and is missing entities.¹⁴ Therefore, we manually confirmed NIL answers and new answers for queries marked as NIL in the data. While an exact comparison is not possible (as described above), our results (94.7%) appear to be at least on par with Cucerzan’s system (91.4% overall accuracy). With the strong results on TAC-KBP, we believe that this is strong confirmation of the effectiveness of our approach.

8 The TAC-KBP Entity Linking Task

We summarize general approaches and results for the 2009 and 2010 TAC-KBP entity linking tasks. A detailed summary can be found in [21].

8.1 Challenging Queries

In their 2009 overview paper McNamee and Dang [30] describe several types of errors prevalent among the most challenging queries.

- Ambiguous acronyms: in query EL1213 (“DRC”) the article refers to the Democratic Republic of Congo as both “DCR” and “DRC”.

¹³ 2008 vs. 2006 version used in [10] We could not get the 2006 version from the author or the Internet.

¹⁴ Since our KB was derived from infoboxes, entities not having an infobox were left out.

Query	Name	DOCID	Entity / KBID
EL2025	Michael Kennedy	NYT_ENG_20010122.0439.LDC2007T07	NIL
Van Brett Watkins, who confessed to the shooting, testified during the trial that Carruth planned to pay him \$5,000 to kill Adams so that Carruth would not have to pay child support. Carruth’s co-defendants, Michael Kennedy , who drove the car Watkins was riding in when he shot Adams, and Stanley Drew Abraham are awaiting trial.			
EL2029	Michael Kennedy	NYT_ENG_19990717.0169.LDC2007T07	E0499939
Michael Kennedy , another of Robert and Ethel’s children, was killed on Dec. 31, 1997, in a bizarre skiing accident. The 39-year-old skied into a tree in Aspen, Colo., while playing a game of ski football.			
EL2030	Michael Kennedy	NYT_ENG_20070430.0025.LDC2009T13	NIL
The Revolution were pinned back and failed to control the ball in counterattacking opportunities in the early going. But things changed quickly as Twellman ran through the halfway line and was taken down by Dax McCarty, who was cautioned by referee Michael Kennedy .			
EL2042	Mike Kennedy	AFP_ENG_20070414.0006.LDC2009T13	NIL
But the Victorian AIDS Council said overseas arrivals accounted for only nine of the 334 new HIV notifications in the state last year and it was wrong to single out immigrants as a source of infection. ”That number is incredibly low,” council president Mike Kennedy told Melbourne’s Age newspaper. ”In Australia the bulk of the epidemic is gay men.”			

Table 5. Sample queries for the entity linking task. Only a small excerpt from the provided document is shown.

- Related organizations: in query EL3871 “Xinhua Finance” is referring to Xinhua Finance Media Ltd., not its parent company Xinhua Finance Ltd.
- Metaphorical names: queries EL1717 and EL1718 (“Iron Lady”) referred to two different women and it wasn’t clear that the nickname was commonly used for them.
- Metonymic references - query EL2599 “New Caledonia” is in a document about World Cup rankings, and it is debatable whether or the the name in the article refers to the country or its national soccer team.
- Assessment errors: queries EL3334 and EL3335 “The Health Department”, are referring to the New York City Department of Health and the New York State Department of Health, respectively, but the United States Department of Health and Human Services was incorrectly judged to be the proper response.

8.2 Approaches

The majority of systems divided the task into three parts: (a) identifying a subset of KB entries that are reasonable candidates for a query entity, and (b) selection of the most likely non-NIL candidate; and (c) deciding whether absence from the KB (*i.e.*, NIL) is the correct response.

Approaches to candidate identification were generally based on name comparisons between query entities and KB entries, often using precomputed dictionary-

ies or inverted files to quickly identify potential KB nodes. A variety of non-exact matching techniques were used, including: alias lists; character n-grams; phonetic matching; acronyms; Wikipedia links or redirects; external Web search; and relationship similarity. Intra-document coreference resolution was exploited by several groups.

A variety of machine learning approaches were used in the selection process. Our submission (HLTCOE [31]) and the QUANTA team [25] used learning to rank frameworks with good effect. In fact, Li et al [25] propose an approach that bears a number of similarities to ours; both systems create candidate sets and then rank possibilities using differing learning methods, but the principal difference is in our approach to NIL prediction. Where we simply consider absence (i.e. the NIL candidate) as another entry to rank, and select the top-ranked option, they use a separate binary classifier to decide whether their top prediction is correct, or whether NIL should be output. We believe relying on features that are designed to inform whether absence is correct is the better alternative. Other approaches for ranking candidates included binary classification and vector comparisons. Ji and Grishman [21] also discuss the TAC-KBP entity linking task in depth and they give a detailed comparison of approaches of different systems in the 2010 evaluation.

9 Beyond Entity Linking

As currently formulated, evaluations of entity linking suffer from a number of limitations. For example, at TAC-KBP, only named mentions (vs. pronouns) have been the linking targets, some names are unresolvable even by humans, and it is challenging to develop query sets to rigorously exercise systems. For example, if a random sample of names is taken, then prominent (and more easily linkable) entities form the majority of the query set; however it is non-trivial to identify challenging queries that contain confusable names of roughly equal prominence.

At present, the only available test sets are in English, although there should be no impediment to developing multilingual and non-English test collections.¹⁵ Another difficulty is the lack of releasable, large-scale knowledge bases. Wikipedia has been the subject of much study, due to its generous licensing, and broad coverage; however, Wikipedia has many unique characteristics (*e.g.*, being indexed by major search engines) that make its use as a test KB susceptible to solutions that do not generalize to other knowledge bases. Further work on Entity Linking in non-English languages will have to deal with case inflections and the similarity metrics which proved most effective in English might not always be optimal. However, the design of our Entity Linking architecture allows most of these components to be pluggable.

¹⁵ As this article went to press we became aware of the efforts by Mayfield *et al.* [27] to construct a cross-language entity linking test collection where the language of the knowledge base is English, but query names are in many languages.

What might the future hold for entity linking? In the near term we expect to see work in multilingual entity linking, increasing interest in linking entity mentions in social media (*e.g.*, Twitter, Facebook), and efforts to increase the diversity and granularity of entity types (*e.g.*, products and brands, events, books, films, and works of art). If Semantic Web technologies continue to gain wider acceptance, then we might expect to see a proliferation of large-scale KBs, which could motivate entity linking beyond its current focus on Wikipedia. Ultimately we expect to see entity linking being used as a component of complex NLP and knowledge discovery applications.

Finally, we should point out that research community has currently split the two problems of cross-document entity coreference and entity linking to a reference knowledge base. Clustering techniques have been dominant in solving the former, while supervised machine learning appears to be the leading approach for the later. It would be beneficial if the research community could better articulate for which real world applications each problem formulation is most applicable, and if possible, develop a unification of these two highly-related problems.

10 Conclusion

We presented a state of the art system to disambiguate entity mentions in text and link them to a knowledge base. Unlike previous approaches, our approach readily ports to KBs other than Wikipedia. We described several important challenges in the entity linking task including handling variations in entity names, ambiguity in entity mentions, and missing entities in the KB, and we showed how to each of these can be addressed. We described a comprehensive feature set to accomplish this task in a supervised setting. Importantly, our method discriminately learns when not to link with high accuracy.

References

1. Artiles, J., Sekine, S., Gonzalo, J.: Web people search: results of the first evaluation and the plan for the second. In: WWW (2008)
2. Asahara, M., Matsumoto, Y.: Japanese named entity extraction with redundant morphological analysis. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1. pp. 8–15. NAACL '03, Association for Computational Linguistics, Stroudsburg, PA, USA (2003), <http://dx.doi.org/10.3115/1073445.1073447>
3. Bagga, A., Baldwin, B.: Entity-based cross-document coreferencing using the vector space model. In: Conference on Computational Linguistics (COLING) (1998)
4. Banko, M., Etzioni, O.: The tradeoffs between open and traditional relation extraction. In: Association for Computational Linguistics (2008)
5. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD Management of Data (2008)

6. Boschee, E., Weischedel, R., Zamanian, A.: Automatic information extraction. In: Conference on Intelligence Analysis (2005)
7. Bunescu, R.C., Pasca, M.: Using encyclopedic knowledge for named entity disambiguation. In: European Chapter of the Association for Computational Linguistics (EACL) (2006)
8. Christen, P.: A comparison of personal name matching: Techniques and practical issues. Tech. Rep. TR-CS-06-02, Australian National University (2006)
9. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. pp. 100–110 (1999)
10. Cucerzan, S.: Large-scale named entity disambiguation based on wikipedia data. In: Empirical Methods in Natural Language Processing (EMNLP) (2007)
11. Cucerzan, S., Yarowsky, D.: Language independent ner using a unified model of internal and contextual evidence. In: proceedings of the 6th conference on Natural language learning - Volume 20. pp. 1–4. COLING-02, Association for Computational Linguistics, Stroudsburg, PA, USA (2002), <http://dx.doi.org/10.3115/1118853.1118860>
12. van Deemter, K., Kibble, R.: On coreferring: Coreference in muc and related annotation schemes. Computational Linguistics 26(4), 629–637 (2000), <http://dblp.uni-trier.de/db/journals/coling/coling26.html#DeemterK00>
13. Dreyer, M., Smith, J., Eisner, J.: Latent-variable modeling of string transductions with finite-state methods. In: Empirical Methods in Natural Language Processing (EMNLP) (2008)
14. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: A survey. IEEE Trans. on Knowl. and Data Eng. 19, 1–16 (January 2007), <http://dx.doi.org/10.1109/TKDE.2007.9>
15. Elsnér, M., Charniak, E.: The same-head heuristic for coreference. In: Association for Computational Linguistics (2010)
16. Fader, A., Soderland, S., Etzioni, O.: Scaling Wikipedia-based named entity disambiguation to arbitrary web text. In: WikiAI09 Workshop at IJCAI 2009 (2009)
17. Gabbard, R., Freedman, M., Weischedel, R.: Coreference for learning to extract relations: Yes virginia, coreference matters. In: Association for Computational Linguistics (2011)
18. Gottipati, S., Jiang, J.: Linking entities to a knowledge base with query expansion. In: Empirical Methods in Natural Language Processing (EMNLP) (2011)
19. Han, X., Sun, L.: A generative entity-mention model for linking entities with knowledge base. In: Association for Computational Linguistics (2011)
20. Huang, J., Treeratpituk, P., Taylor, S., Giles, C.L.: Enhancing cross document coreference of web documents with context similarity and very large scale text categorization. In: Conference on Computational Linguistics (COLING) (2010)
21. Ji, H., Grishman, R.: Knowledge base population: Successful approaches and challenges. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-HLT) (2011)
22. Joachims, T.: Optimizing search engines using clickthrough data. In: Knowledge Discovery and Data Mining (KDD) (2002)
23. Klein, M., Nelson, M.L.: A comparison of techniques for estimating IDF values to generate lexical signatures for the web. In: Workshop on Web Information and Data Management (WIDM) (2008)
24. Lehmann, J., Monahan, S., Nezda, L., Jung, A., Shi, Y.: Lcc approaches to knowledge base population at tac 2010. In: Proc. TAC 2010 Workshop (2010)

25. Li, F., Zhang, Z., Bu, F., Tang, Y., Zhu, X., Huang, M.: THU QUANTA at TAC 2009 KBP and RTE track. In: Text Analysis Conference (TAC) (2009)
26. Mann, G.S., Yarowsky, D.: Unsupervised personal name disambiguation. In: Conference on Natural Language Learning (CONLL) (2003)
27. Mayfield, J., Lawrie, D., McNamee, P., Oard, D.W.: Building a cross-language entity linking collection in twenty-one languages. In: Proceedings of the Cross Language Evaluate Forum (CLEF) (2011)
28. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4. pp. 188–191. CONLL '03, Association for Computational Linguistics, Stroudsburg, PA, USA (2003), <http://dx.doi.org/10.3115/1119176.1119206>
29. McCallum, A., Nigam, K., Ungar, L.: Efficient clustering of high-dimensional data sets with application to reference matching. In: Knowledge Discovery and Data Mining (KDD) (2000)
30. McNamee, P., Dang, H.T.: Overview of the TAC 2009 knowledge base population track. In: Text Analysis Conference (TAC) (2009)
31. McNamee, P., Dredze, M., Gerber, A., Garera, N., Finin, T., Mayfield, J., Piatko, C., Rao, D., Yarowsky, D., Dreyer, M.: HLT/COE approaches to knowledge base population at TAC 2009. In: Text Analysis Conference (TAC) (2009)
32. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. *Linguisticae Investigationes* 30(1), 3–26 (January 2007), <http://nlp.cs.nyu.edu/sekine/papers/li07.pdf>, publisher: John Benjamins Publishing Company
33. Ng, V.: Supervised noun phrase coreference research: The first fifteen years. In: Proceedings of the ACL. pp. 1396–1411 (2010)
34. Ng, V.: Supervised noun phrase coreference research: The first fifteen years. In: Association for Computational Linguistics (2010)
35. Poesio, M., Day, D., Artstein, R., Duncan, J., Eidelman, V., Giuliano, C., Hall, R., Hitzeman, J., Jern, A., Kabadjov, M., Yong, S., Keong, W., Mann, G., Moschitti, A., Ponzetto, S., Smith, J., Steinberger, J., Strube, M., Su, J., Versley, Y., Yang, X., Wick, M.: Exploiting lexical and encyclopedic resources for entity disambiguation: Final report. Tech. rep., JHU CLSP 2007 Summer Workshop (2008)
36. Popescu, O.: Dynamic parameters for cross document coreference. In: Conference on Computational Linguistics (COLING) (2010)
37. Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., Manning, C.: A multi-pass sieve for coreference resolution. In: Empirical Methods in Natural Language Processing (EMNLP) (2010)
38. Rao, D., McNamee, P., Dredze, M.: Streaming cross document entity coreference resolution. In: Conference on Computational Linguistics (COLING) (2010)
39. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill Book Company (1983)
40. Sang, E.T.K., Meulder, F.D.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. In: Conference on Natural Language Learning (CONLL) (2003)
41. Simpson, H., Parker, R., Strassel, S., Dang, H.T., McNamee, P.: Wikipedia and the web of confusable entities: Experience from entity profile creation for tac knowledge base population. In: Proceedings of the Seventh International Language Resources and Evaluation Conference (LREC) (2010)
42. Singh, S., Subramanya, A., Pereira, F., McCallum, A.: Large-scale cross-document coreference using distributed inference and hierarchical models. In: Association for Computational Linguistics (2011)

43. Stoyanov, V., Cardie, C., Gilbert, N., Riloff, E., Buttlar, D., Hysom, D.: Reconcile: A coreference resolution research platform. In: Association for Computational Linguistics (2010)
44. Sutton, C., McCallum, A.: Introduction to conditional random fields for relational learning. In: Getoor, L., Taskar, B. (eds.) Introduction to Statistical Relational Learning. MIT Press (2006)
45. Syed, Z., Finin, T., Joshi, A.: Wikipedia as an ontology for describing documents. In: Proceedings of the Second International Conference on Weblogs and Social Media. AAAI Press (2008)
46. Yang, X., Zhou, G., Su, J., Tan, C.L.: Coreference resolution using competition learning approach. In: In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. pp. 176–183 (2003)
47. Zhang, W., Sim, Y.C., Su, J., Tan, C.L.: Entity linking with effective acronym expansion instance selection and topic modeling. In: International Joint Conference on Artificial Intelligence (2011)
48. Zhang, W., Sim, Y., Su, J., Tan, C.: Nus-i2r: Learning a combined system for entity linking. In: Proc. TAC 2010 Workshop (2010)
49. Zhang, W., Su, J., Tan, C.L.: Entity linking leveraging automatically generated annotation. In: Conference on Computational Linguistics (COLING) (2010)