DAMSEL: The DSTO/Macquarie System for Entity-Linking

Matthew Honnibal
Centre for Language Technology
Macquarie University, Australia
matthewh@science.mq.edu.au

Robert Dale
Centre for Language Technology
Macquarie University, Australia
rdale@science.mq.edu.au

Abstract

This paper describes the DSTO/Macquarie University System for Entity Linking (DAMSEL), which competed in the 2009 Text Acquisition Conference Knowledge Base Population task. The system achieves 73.5% accuracy.

For a given named entity mention, the system selects a set of candidate entities from the knowledge base and selects the most likely candidate based on the similarity between the document in which the mention was found and the candidate's Wikipedia article. The best-performing candidate selection strategy took advantage of Wikipedia redirection and disambiguation data. The best-performing similarity measure was the cosine metric.

1 Task Overview

Strings that refer to named entities are often ambiguous, and the same entity can be referred to by many different strings. This means information extraction systems can benefit from systems which resolve mentions to an unambiguous node in a knowledge base, instead of assuming that each named entity string refers to a unique entity. The simplest way to acquire a high coverage knowledge base is to use Wikipedia, as its markup and link structure allows a lot of useful information to be extracted.

The first system to resolve entity mentions to Wikipedia pages was described by Bunescu and Pasca (2006). The system was trained and evaluated using Wikipedia links, a methodology followed by Mihalcea and Csomai (2007). The idea is that all incoming links to an entity's Wikipedia article are unambiguous mentions of that entity. Wikipedia authors can anchor these links with arbitrary text, allowing them to realise the entity mention however they like. However, the entity links are still subject to the specific genre conventions of Wikipedia, so may not represent the full diversity of how entities are mentioned in other text types, making this a problematic evaluation measure.

The TAC-KBP challenge corrects this by providing a manually annotated evaluation set of entities in news text, linked to entities in a knowledge base extracted from Wikipedia. The entity linking task presented users with a mention string and the ID of the document it was drawn from. The task was to select the node in the knowledge base that the entity referred to. If there was no corresponding node in the knowledge base, systems were required to return *nil*. We did not participate in the second phase of the task, where properties in the knowledge base would be populated from the linked document.

2 System Design

We divide the task of entity linking into two phases. During the *candidate selection* phase, the mention string is used to retrieve a set of candidate entities from the knowledge base. During the *similarity measuring* phase, we calculate the similarity of each candidate entity's Wikipedia page with the document containing the mention string. The entity with the most similar document is returned, so long as the similarity is above a given threshold.

Dictionary	True_{entity}	True_{nil}	False_{entity}	False_{nil}	$\lvert Candidates \rvert$
Truncated name	4.7	56.7	23.7	15.0	2.3
Disambiguation	12.3	51.5	16.1	20.1	5.3
Link anchors	10.6	46.8	16.8	25.8	14.5
Union	16.5	43.6	11.8	28.1	17.9

Table 1: Accuracy and ambiguity rates for candidate sets returned by Wikipedia dictionaries. A candidate set is judged True if it returns the correct entity (or *nil*).

We used the Wikipedia pages provided as part of the TAC-distributed knowledge base. We did not use any of the other information in the knowledge base, such as the entity type or the facts extracted from the infoboxes.

2.1 Candidate Selection

We experimented with two candidate selection methods. One attempts to select a minimal set of entities, to minimise ambiguity; the other selects a more inclusive set of entities, to maximise coverage.

2.1.1 The Minimal Ambiguity Strategy

The *minimal ambiguity* strategy uses a series of look-ups, ordered according to their reliability. The first resource we consult is a dictionary of Wikipedia page names. Page names are unique in Wikipedia, so this can return at most one entity. An entity is returned for 19% of development set queries. Of these, the 86% were correct. The answer was *nil* for 6% of queries, and a different entity for 8%.

The next dictionary that we use is drawn from Wikipedia's redirection data. Wikipedia contains pages that simply redirect to other articles, effectively providing synonymy sets. Each redirection page can only point to a single Wikipedia page, so this dictionary is also limited to returning one or zero entities. The redirection dictionary has similar coverage to the name dictionary (19%), although it is less accurate. 59% of the entities it returns are correct. The errors were quite evenly split between the two possible cases: for 19% of the queries, the answer was an entity other than the one returned; for 23%, the answer was *nil*.

The remaining dictionaries can potentially return more than one entity. The first such dictionary is built by *name truncation.* Many Wikipedia page titles contain an appositive or parenthetical part, such as *Alabama* and *band* in, respectively, *Birmingham, Alabama* and *Garbage (band).* We form a dictionary keyed by the first part of the page title (identified by all text up to a comma or open parenthesis), the values of which are collections of entities with titles like *Birmingham, Alabama* and *Birmingham, Michigan*. Similar information can be found in Wikipedia disambiguation pages. For instance, the disambiguation page titled *Birmingham* includes links to 41 different entities that *Birmingham* might refer to. Finally, we can find a similar source of information in the text used to anchor links between pages. For instance, the *Alabama* page might have a link to *Birmingham, Alabama* anchored by the text *Birmingham.* We therefore compile a dictionary mapping anchor texts to the entities they refer to.

Table 1 summarises the performance of these dictionaries. The $|Candidates|$ column shows the mean cardinality of non-empty candidate sets retrieved by the dictionary. Alarmingly, none of these resources return a high rate of true positives. This means that a disambiguation algorithm will have to perform very well to make using these resources worthwhile. The *Union* row shows the figures for a dictionary built from all three resources.

The *false nil* case will be especially hard to account for. In these cases, one or more candidates are returned for a query whose answer is *nil.* This means that the disambiguator must either include a special model to predict *nil* values, or employ a similarity threshold below which an entity is assigned *nil*. For each dictionary, a baseline of always assigning *nil* would outperform a classifier that assumed that one candidate must be assigned, because the number of queries where the answer is *nil* and a candidate is returned (the True_{nil} column of Table 1) is lower than the number of queries where the correct entity is among the candidates returned.

2.1.2 The High Coverage Strategy

The *high coverage* strategy involves simply looking up the mention string in a set of reverse indexes extracted from the Wikipedia mark up. The set consists of the *name*, *redirection*, *truncated name* and *disambiguation* dictionaries.

We selected this set of dictionaries empirically, using the TAC development data. We included systems using this strategy because we were concerned that the *nil* rate in the test data might be much lower than the *nil* rate in the development data, due to the different collection strategies used. The task documentation reported that the development data was chosen largely at random, while the evaluation data was selected with a bias towards more ambiguous and interesting cases. A lower *nil* rate might make our minimal ambiguity strategy inappropriate, because it only returns the entities found in the most reliable dictionaries.

We did not, however, include the link anchor text dictionary in the high coverage strategy. We based this decision on our observation that this dictionary returned a very high number of candidates. Given that our disambiguation strategies were not very powerful, it seemed likely that this dictionary would hinder performance, even on data with a much lower *nil* rate.

2.2 Similarity Measures

We experimented with two similarity measures: the cosine similarity measure, and a simple measure, which we call *token overlap*, that simply measures the cardinality of the intersection between the two sets of tokens. Both measures operate on a bag-of-words extracted from the Wikipedia article, and the mention's document.

We performed a few experiments on the TAC-KBP development data, which contained 193 annotated mentions. We found that stemming, stopping and case normalisation had little effect on our results. We also experimented with a simple extractive summarisation system, which consisted of removing all sentences that did not contain the mention string. This slightly improved performance.

The cosine similarity between the mention's context and the candidate's context is the sum of the weighted product of each term that occurs in both contexts:

$$\mathit{Sim}(c,m) = \sum_{\mathit{common\ terms}\ t_j} wc_j \times wm_j \quad (1)$$

where t_j is a term present in c and m, wc_j is its weight in c and wm_j is its weight in m. The weight of a document d is computed using TF-IDF and cosine normalisation:

$$wd_j = \frac{tf \times \log \frac{N}{df}}{\sqrt{w_{d1}^2 + w_{d2}^2 + ... + w_{dn}^2}} \quad (2)$$

where tf is the frequency of the term t_j in the document d, N is the total number of documents in the collection being examined, and df is the number of documents in the collection that the term t_j occurs in. The denominator is the *cosine normalization factor*.

For the cosine similarity measure, we applied a minimal similarity threshold of 0.1. If the maximum similarity measure was below this threshold, the system returned *nil*. The threshold was determined empirically on the development data. A threshold of 10 was used for the token overlap system.

3 Results

We experimented with all combinations of our two candidate selection and disambiguation systems. The results are shown in Table 2. Both systems using the cosine similarity measure performed better than the simpler token overlap strategy, which was effectively an unnormalised version of the Jaccard similarity coefficient. Given the simplicity of the token overlap measure, and the fact that the cosine metric is well established in NLP and was shown to be effective at cross-document coreference resolution by Bagga and Baldwin (1998), this result is unsurprising.

The best candidate selection strategy was the *minimal ambiguity* method, which looked up the mention string in a series of dictionaries, and returned the candidates as soon as at least one match was found. The problem with the *high coverage*

Candidate Selection	Disambiguation	In KB	Nil	All
High coverage	Token Overlap	52.3	51.0	51.5
High coverage	Cosine	66.9	59.8	62.3
Low ambiguity	Token Overlap	63.0	69.0	66.5
Low ambiguity	Cosine	64.9	79.9	73.5
Highest accuracy of TAC systems		77.3	83.5	82.2
Median accuracy of TAC systems		63.5	78.9	71.1
nil baseline		0.0	1.0	67.5

Table 2: Comparison of our candidate selection and disambiguation strategies on the TAC-KBP entity linking evaluation set.

strategy may be that the *nil* case was quite common in the data, consisting of 67.5% of the entries. This provided a strong advantage for the minimal ambiguity method, which returned either a small set of entities, or a prediction of *nil*.

The cosine similarity and minimal ambiguity system was the only one to outperform the *nil* baseline. The system performed slightly above the reported median for the TAC participants, but substantially below the best reported accuracy.

4 Conclusion

Entity linking systems, such as those described by Bunescu and Pasca (2006) and Cucerzan (2007), are fairly complex. We were not aware of any published results that investigated how difficult the task was by evaluating a simple system using fairly general techniques. We therefore decided to enter a minimal system in the TAC-KBP entity linking challenge to see how this would perform.

The best performing configuration, using a minimal ambiguity candidate selection and the cosine measure, achieved performance slightly above the median of the systems entered in the challenge. However, the accuracy of 73.5% still leaves a lot of room for improvement, and is substantially below the best performing system, which reportedly achieved 82.2%. This suggests that entity linking is a difficult task, and is not easily solved using a fairly trivial system.

References

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 17th International Conference on Computational Linguistics*, pages 79–85. Association for Computational Linguistics, Montreal, Canada.

Razvan Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 9–16. Association for Computational Linguistics, Trento, Italy.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716. Association for Computational Linguistics, Prague, Czech Republic.

Rada Mihalcea and Andras Csomai. 2007. Wikify: linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007.*