Document-level Entity Linking: CMCRC at TAC 2010

Will Radford[†‡] **Ben Hachey**[‡◇] **Joel Nothman**[†‡] **Matthew Honnibal**[†‡] **James R. Curran**[†‡]

[†]School of Information Technologies
University of Sydney
NSW 2006, Australia

[‡]Capital Markets CRC
55 Harrington Street
NSW 2000, Australia

[◇]Centre for Language Technology
Macquarie University
NSW 2109, Australia

{wradford,joel,mhonn,james}@it.usyd.edu.au bhachey@cmcrc.com

Abstract

This paper describes the CMCRC systems entered in the TAC 2010 entity linking challenge. The best performing system we describe implements the document-level entity linking system from Cucerzan (2007), with several additions that exploit global information. Our implementation of Cucerzan's method achieved a score of 74.9% in development experiments. Additional global information improves performance to 78.4%. On the TAC 2010 test data, our best system achieves a score of 84.4%, which is second in the overall rankings of submitted systems.

1 Introduction

Named Entity Linking (NEL) is a practical extension of Named Entity Recognition where named entity mentions are grounded to the entities to which they refer. The TAC NEL component of the Knowledge Base Population track frames the task as follows. Each query (consisting of a mention term and a document in which it appears) should be linked to a knowledge base (KB) entry or NIL if the term refers to an entity *outside* the KB. The TAC data uses news and web data as context "source documents" and the KB is derived from English Wikipedia pages. The key challenge is disambiguation: there are 26 possible `John Howard` Wikipedia entries and a query may refer to one of these or a less notable entity.

The TAC 2009 competition yielded a variety of approaches to NEL. Varma et al. (2009) use a Wikipedia snapshot to generate an alias repository of redirects, disambiguation pages and bold text from the first paragraph. They then backoff over Lucene[1] indices to generate a list of candidate entities and use Cosine similarity between entity text and source document to choose the best link. Since an entire Wikipedia snapshot is used, a NIL link is returned if the best link is in the index but *not* in the KB.

The second and third systems from TAC 2009 both use learning to rank approaches. Li et al. (2009) use ListNet with a feature set that includes attributes based on the similarity between the query mention and the entity, including named entity overlap. McNamee et al. (2009) use SVM$^{\text{rank}}$ with a diverse feature set that includes Cosine similarity, search engine popularity measures and named entity counts.

Overall, few of the TAC 2009 systems use information beyond the local context for a given query mention. We believe that NEL will benefit from global information (e.g., from the *whole* document, from all of Wikipedia). To incorporate this, we follow the global entity disambiguation approaches proposed by Cucerzan (2007) to link all entities found in the source document. We also use document-level coreference resolution, alias reliability information and Wikipedia graph structure.

Development experiments demonstrate that all three sources of global information are useful, improving performance from 74.9% to 78.4%. We also present updated results on the TAC 2010 test data, where a system using these information sources together achieves a score of 84.4%. This ranks second overall and is competitive with the best non-web submission (85.8%).

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

$\mathcal{Q}$	The set of queries
m_q	The mention string for query q ($q \in \mathcal{Q}$)
d_q	The document for query q ($q \in \mathcal{Q}$)
e_q	System output for query q (KB ID or NIL)
g_q	The gold standard KB ID for q (or NIL)

Table 1: TAC data specification.

2 TAC 2010 Entity Linking

2.1 Knowledge Base

TAC entity linking is performed with respect to an entity KB derived from Wikipedia. Each entity in the KB (sometimes called a node) includes 1) a name string (e.g., Bud Abbot), 2) a KB node ID (e.g., E0064214), and 3) the text from the corresponding Wikipedia page. Entity nodes also include fields for entity type (i.e., person - PER, organisation - ORG, geo-political - GPE, or unknown) and fields specific to each entity type that are derived from Wikipedia infoboxes. However, these are not used in the system described here.

The TAC KB is derived from pages in the October 2008 Wikipedia dump 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.

2.2 Task

Table 1 contains definitions for the TAC entity linking task. The input is the set of queries $\mathcal{Q}$. Each query $q \in \mathcal{Q}$ includes a mention string m_q and a document d_q in which m_q is found. For example, for query EL11 in the TAC 2009 test data, m_q = Abbot and d_q is a news article with the title New Releases: Coming Oct. 28 for which the 5th paragraph starts with the text Also on DVD Oct. 28: ``Abbot and Costello: The Complete Universal Pictures Collection''.

The goal is to automatically find the node in the KB that corresponds to the entity referred to by m_q in document d_q. For instance, the query mention Abbot in the example above refers to the actor for whom there is a KB node with IDE0064214 and name string Bud_Abbot.

The gold standard annotation g_q consists of the gold standard KB ID. Queries can also have no corresponding entry in the KB, in which case g_q = NIL. Overall accuracy is measured by the proportion of queries that were correctly linked:

$$A = \frac{|\{q|e_q = g_q\}|}{|\mathcal{Q}|} \tag{1}$$

Accuracy is also measured separately for the subset of queries where $g_q \neq$ NIL ($A_{\mathcal{C}}$) and for the subset of queries where g_q = NIL ($A_{\emptyset}$).

	TAC 2009 test		TAC 2010 train		TAC 2010 test	
$\lvert\mathcal{Q}\rvert$	3,904		1,500		2,250	
KB	1,675	(43%)	1,074	(72%)	1,020	(45%)
NIL	2,229	(57%)	426	(28%)	1,230	(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%)

Table 2: Comparison of TAC data sets.

2.3 Data Sets

We use the TAC 2010 training data and the TAC 2009 test data for system development. These are summarised in Table 2, as is the TAC 2010 test data. The first difference between data sets is in terms of the proportion of NIL queries. 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 terms of 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 web documents as well. The TAC 2010 training data is roughly evenly divided between news and web documents, while the test data is skewed towards news.

As we did not know the proportion of NIL queries before the TAC 2010 test data was released, we preferred systems that achieved more balanced accuracy across NIL and non-NIL queries. We did not consider performance across entity types or source genres during development.

3 Wikipedia Infrastructure

Wikipedia Wikipedia has several features that make it useful for the NEL task. Articles are a key feature and contain information about a particular topic (perhaps an entity), minimally featuring a title and text. Text may contain links to Wikipedia or other web pages and belong to different categories, comprising a large graph that describes how articles relate to one another. Redirect and disambiguation pages are specifically useful to NEL. Redirect pages map titles to the canonical article title, providing a list of aliases for an article. As an encyclopedia, Wikipedia emphasises disambiguation, and disambiguation pages provide extra text to disambiguate articles with similar titles.

Wikipedia Processing Infrastructure We use the November 2009 dump of `http://en.wikipedia.org` (without revision history), which contains in the order of 3.3M articles and is 11.8GB of `bzip`-compressed XML. We process the dump files and build key-value stores using Tokyo Tyrant[2] that store the article content. This allows quick access to articles by title as well as the ability to stream through all articles. The article content uses MediaWiki markup, a powerful system that allows inclusion of arbitrary HTML and templates that must be expanded before the content can be parsed. We use the *mwlib* parser[3] to extract article text, categories, links, disambiguation and redirect information. These are stored in the cabinet to allow convenient access and processing.

Text Indices for Candidate Generation We index the Wikipedia dump using the Solr search server. Each document in the index corresponds to a Wikipedia page. Index fields include article text as well as the title and various alias fields. Aliases are derived from 1) titles of incoming redirect pages, 2) titles of incoming disambiguation pages 3) bold words from the first paragraph of an article, and 3) the anchor text of incoming wiki links that occur at least twice. Alias fields are specified as multi-valued and each alias instance is added. Aliases with parentheses (e.g., `Texas (band)`) and commas (e.g., `Sydney, Nova Scotia`) are considered to have apposition phrases. For these, two aliases are generated. The first includes the entire string and the second includes just the portion before the apposition (e.g., `Texas`, `Sydney`).

4 Baseline System

Our baseline system uses a conservative, back-off based strategy to retrieve entity candidates from the Wikipedia dump, using alias sources from Wikipedia. Once a set of candidates has been found, they are ranked according to the Cosine similarity of their Wikipedia pages and the source document. The entity corresponding to the highest ranked article is then returned — or NIL, if the article has no corresponding entity in the KB. The system is similar to the DAMSEL system entered in the TAC competition (Honnibal and Dale, 2009).

Because the Cosine measure is not a powerful disambiguator, the candidate generation strategy is tuned towards precision, rather than recall. This is achieved by ordering the alias sources according to their reliability, and stopping once an alias source is found to return at least one candidate. If no candidates are returned, the next alias source is consulted. The alias sources used are:

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

It is important for the system's performance that these alias sources are consulted one-by-one. This prevents a candidate generated by a less reliable alias source from being ranked ahead of an article from a more reliable alias, such as title or redirect. A threshold of Cosine similarity 0.01 is applied for all alias sources past the first, so an article must either have a title that literally mentions the query, or have page text that is minimally similar to the source document. The order of the alias sources and the Cosine threshold were determined experimentally on the TAC 2009 data.

[2] `http://fallabs.com/tokyotyrant/`

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

$\mathcal{M}_q$	Entity mentions in d_q (including m_q)
$\mathcal{E}_m$	Candidate entities for mention m
$\mathcal{C}_e$	Categories for entity e ($e \in \mathcal{E}_m$)
$\mathbf{c}$	Document-level category vector
$\mathbf{c}_c$	$\sum_{m \in \mathcal{M}_q} \lvert\{e \mid e \in \mathcal{E}_m \wedge c \in \mathcal{C}_e\}\rvert$
$\mathcal{T}_e$	Contexts for entity e ($e \in \mathcal{E}_m$)
$\mathbf{t}$	Document-level context vector
$\mathbf{t}_t$	Frequency of context t in document d_q

Table 3: Definitions for Cucerzan-style disambiguation using document-level vectors.

5 Core System: Cucerzan Implementation

This section describes our implementation of Cucerzan's (2007) document-level NE linker.

Resources Category and context information is extracted for each Wikipedia entity page. Categories include the Wikipedia categories. These are filtered to remove meta-categories or those deemed not useful for disambiguation. We exclude categories if their name: contains a stop word (`article`, `page`, `date`, `year`, `birth`, `death`, `living`, `century`, `acronym`, `stub`); contains a four-digit number (i.e., a year); or is `Exclude in print`. Categories for a page also include the title of list and table pages that link to it. List and table pages are identified by looking for page titles that start with `List of` and `Table of` respectively.

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

Candidate Generation We first use the tokeniser and NER tagger from the C&C Tools (Curran et al., 2007) to tokenise and extract named entity mentions from the text. Each mention is checked against the KB to see if it matches an article title exactly. If it does not, candidate lists are generated using an exact match lookup against the Solr Wikipedia index on the following fields: article titles, redirect titles, disambiguation titles, redirect titles for disambiguation pages, and bold terms in disambiguation pages.

Candidate Disambiguation Cucerzan disambiguates the query mention with respect to document-level vectors derived from all entity mentions. Document-level vectors are calculated for Wikipedia categories and for contexts as defined in Table 3. The value of the entry for a category c in the document-level category vector $\mathbf{c}$ is the sum across mentions of candidate entities that have c assigned ($\sum_{m \in \mathcal{M}_q} \lvert\{e \mid e \in \mathcal{E}_m \wedge c \in \mathcal{C}_e\}\rvert$). And, the value of the entry for a context t in the document-level context vector $\mathbf{t}$ is the frequency of context t in the text of document d_q. Given these definitions, a candidate entity e is scored as follows:

$$s_{Cucerzan}(e) = \Big(\sum_{c \in \mathcal{C}_e} \mathbf{c}_c + \sum_{t \in \mathcal{T}_e} \mathbf{t}_t\Big) - \lvert\mathcal{C}_e\rvert \qquad (2)$$

The first term ($\sum_{c \in \mathcal{C}_e} \mathbf{c}_c$) assigns a score to a candidate entity e based on the popularity of its categories across candidates for all entity mentions in the document. This is equivalent to the scalar product between the document-level category vector $\mathbf{c}$ and a 01 entity-level category vector (i.e., $\langle 1 \text{ if } c \in \mathcal{C}_e, \text{ else } 0\rangle_c$). The second term ($\sum_{t \in \mathcal{T}_e} \mathbf{t}_t$) assigns a score to e based on the document-level popularity of its contexts. This is equivalent to the scalar product between the document-level context vector and a 01 entity-level context vector. Finally, the third term ($\lvert\mathcal{C}_e\rvert$) is subtracted out to avoid rewarding entities for being similar to their own contribution to the document-level vector and to avoid rewarding candidates for simply having long Wikipedia pages that are members of many categories. The best candidate for a given mention m is the argmax of $s(e)$ over candidates $e \in \mathcal{E}_m$:

$$d(m) = \operatorname*{argmax}_{e \in \mathcal{E}_m} s(e). \qquad (3)$$

Replication Our implementation scores 86.8% on Cucerzan's news data while he reports 91.4%. The main differences in our implementation are as follows: 1) we use a different NER system; 2) we do not include incoming link anchor text as an alias source; 3) we may use a different set of categories due to category name and Wikipedia change; and 4) we do not shrink source document context where no clear entity candidate can be identified for a mention.

NIL Handling NIL is returned if there are no candidates ($\mathcal{E}_m = \emptyset$) or if the top candidate can not be mapped to a node in the KB. The second criterion performs NIL detection through linking, exploiting the full Wikipedia dump. This is comparable to the approach in Varma et al. (2009).

		TAC 2010 train			TAC 2009 test		
Method	Sys ID	A_C	$A_\emptyset$	A	A_C	$A_\emptyset$	A
Cosine	CMCRC 2	80.6	85.9	82.1	64.1	83.4	75.1
Cucerzan Replication		88.0	81.2	86.1	70.7	78.1	74.9
+ Coreference		88.5	81.2	86.4	72.4	80.9	77.3
+ Reliability		89.0	82.2	87.1	71.2	78.7	75.5
+ Graph		88.6	81.7	86.7	72.8	78.5	76.0
+ Coref+Rel		89.2	82.2	**87.2**	72.6	81.6	77.7
+ Coref+Graph	CMCRC 3	89.0	82.2	87.1	74.2	81.5	**78.4**
+ Rel+Graph		89.0	82.2	87.1	71.2	78.7	75.5
+ Coref+Rel+Graph	CMCRC 1	89.2	82.2	**87.2**	72.6	81.6	77.7

Table 4: Development results on TAC 2010 training data and TAC 2009 test data.

6 Extensions to Core System

Coreference Handling Naïve in-document coreference is performed by taking each mention and trying to match it to a previous mention in the document. A match is made if the mention exactly matches a previous mention, or is a right-aligned token subsequence of a previous longer mention. For example the mention `Howard` will be co-referred to a previous mention `John Howard`. The longest form in a coreference chain is considered to be the most canonical and is used as an expanded search term for an input query. This is comparable to the approach in Cucerzan (2007).

Alias Reliability The collaborative editing approach in Wikipedia makes it a noisy source of entity aliases. We apply a filter to remove noisy alias, particularly from link anchor text and disambiguation pages. We group aliases by normalising case and punctuation, then define an alias as reliable if it corresponds to a title, redirect, or bold term; if it is known from link text and another source; if it forms the initials of another reliable aliases of three or more words; or if it contains at least 50% of the words of another reliable multi-word alias. Our system then requires, that a mention's candidate entities have the mention as a reliable alias.

Graph-based Reweighting A second disambiguator based on the Wikipedia link structure is also applied to the ranked candidate lists. This examines the top candidate for each mention and calculates the union of all in-links into those entities. Then, for each candidate of each mention, the score is re-weighted using a logged size of the intersection of its in-links with the set of global in-links. The weight for each candidate entity is calculated as

$$s_{Graph}(e) = \log(|\mathcal{L}_e \cap \mathcal{L}_q| + 1) + 1 \qquad (4)$$

where $\mathcal{L}_e$ is the set of reciprocal links for entity e and $\mathcal{L}_q$ is the combined set of in-links from the top candidates for all other mentions in d_q. s_{Graph} is used to re-weight $s_{Cucerzan}$ by multiplication.[4]

7 Development Experiments

Table 4 contains development experiments exploring the parameter space of our extended Cucerzan system. We report results on the TAC 2010 training data and on the TAC 2009 test data. The first row corresponds to the baseline system from Section 4, which is also one of our submissions to the official evaluation (CMCRC 2). The next row corresponds to our replication of the Cucerzan document-level disambiguation approach from Section 5. And, all following rows correspond to different combinations of the extensions described in Section 6.

The second through fifth result rows show that all extensions make a very similar positive contribution, improving scores on the TAC 2010 training data from 86.1% by 0.3 to 1 point in overall accuracy. On the TAC 2009 test data, the improvement is more substantial for coreference and graph-based reweighting, which improve overall accuracy by 2.4 and 1.1 points respectively. Combining these extensions improves results further, leading to an improvement of as much as 3.5 points in overall accuracy. All three extensions improve A_C, with graph-based reweighting having a the strongest effect.

[4] s_{Graph} will have a minimum of 1 if there is an empty set and will thus result in no reduction of score when multiplied.

	Official			Updated		
System	$A_{\mathcal{C}}$	$A_{\emptyset}$	A	$A_{\mathcal{C}}$	$A_{\emptyset}$	A
CMCRC 2	61.0	91.6	77.7	61.1	92.9	78.5
CMCRC 3	73.7	88.7	**81.9**	78.4	89.1	84.3
CMCRC 1	69.0	90.8	80.9	79.0	88.8	**84.4**

Table 5: TAC 2010 test results.

	News			Web		
Type	ORG	GPE	PER	ORG	GPE	PER
CMCRC 2	72.0	62.2	96.4	85.3	83.2	85.6
CMCRC 3	78.8	80.2	97.2	83.2	74.3	88.4
CMCRC 1	79.6	79.6	97.0	84.0	74.7	88.0

Table 6: TAC 2010 test results by genre and entity type.

8 Results

Table 5 contains our overall results on the TAC 2010 test data. The first three result columns contain our scores from the official TAC evaluation. This used an early version of our infrastructure, which had tokenisation and an alias reliability implementations that resulted in slightly worse accuracy. The official submission also used an early version of our Cucerzan implementation. This only allowed anchor text context when the link was reciprocal **and** appeared in the first paragraph whereas Cucerzan (and our updated results) take the union. Finally, we added acronym handling. For example, UN would now be coreferred with any corresponding long forms earlier in the document (e.g., `United Nations`).

Overall, the test results follow the same trend as our development experiments. The baseline system (CMCRC 2) achieves an overall accuracy score of 78.5% while the Cucerzan replication with coreference handling and graph-based reweighting (CMCRC 3) achieves a substantially higher overall accuracy of 84.3%. And the addition of reliability filtering (CMCRC 1) brings the overall accuracy of 84.4%. With respect other TAC 2010 submissions, our updated CMCRC 1 system ranks second (compared to a maximum scores of 86.8% overall and 85.8% for systems that do not access the web during run time). Our systems all perform better on NIL queries than on KB queries though the updated implementation of alias reliability seems to help balance KB and NIL accuracy. This increases KB accuracy 1.6 points with an insignificant drop in NIL accuracy of 0.3.

9 Analysis

Table 6 contains a breakdown of scores by source genre (News, Web) and entity type (ORG, GPE, PER). Both systems based on our Cucerzan replication (CMCRC 3 and CMCRC 1) perform best on PER entities. These do particularly well on the web data where they outperform the baseline system (CMCRC 2) by 6 points in overall accuracy. These systems also do particularly well on GPE entities in the news domain, outperforming baseline by 13 to 15 points. For all other columns, there is much less variation in scores with a maximum difference of approximately 2 points accuracy. The baseline system achieves an accuracy approximately 1 point higher than the other systems on GPE entities in the web domain.

References

Silviu Cucerzan. 2007. 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*, pages 708–716, Prague, Czech Republic.

James Curran, Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion*, pages 33–36, Prague, Czech Republic.

Matthew Honnibal and Robert Dale. 2009. DAMSEL: The DSTO/Macquarie system for entity-linking. In *Proceedings of the Text Analysis Conference*, Gaithersburg, MD, USA.

Fangtao Li, Zhicheng Zheng, Fan Bu, Yang Tang, Xiaoyan Zhu, and Minlie Huang. 2009. THU QUANTA at TAC 2009 KBP and RTE track. In *Proceedings the Text Analysis Conference*, Gaithersburg, MD, USA.

Paul McNamee, Mark Dredze, Adam Gerber, Nikesh Garera, Tim Finin, James Mayfield, Christine Piatko, Delip Rao, David Yarowsky, and Markus Dreyer. 2009. HLTCOE approaches to knowledge base population at TAC 2009. In *Proceedings the Text Analysis Conference*, Gaithersburg, MD, USA.

Vasudeva Varma, Praveen Bysani, Kranthi Reddy, Vijay Bharat, Santosh GSK, Karuna Kumar, Sudheer Kovelamudi, Kiran Kumar N, and Nitin Maganti. 2009. IIIT Hyderabad at TAC 2009. In *Proceedings the Text Analysis Conference*, Gaithersburg, MD, USA.