Stanford-UBC Entity Linking at KBP TAC 2010

ANGEL X. CHANG, VALENTIN I. SPITKOVSKY, ERIC YEH
ENEKO AGIRRE, CHRISTOPHER D. MANNING
{angelx,vals,yeh1}@stanford.edu
e.agirre@ehu.es, manning@stanford.edu
IXA NLP Group, University of the Basque Country and Computer Science Department, Stanford University

Entity Linking

Principles of our Entity Linking system:
- Simple system
- Context indepedent dictionary that maps entity mentions to candidate Wikipedia articles
- Context dependent heuristic disambiguation using NER and coreference
- NIL as matches of entities in Wikipedia outside the KB

Previously, for TAC-KBP 2009 we developed two other disambiguation components
- A supervised disambiguation system, based on Word Sense Disambiguation techniques.
- A knowledge-based system, using overlap with text of KB and Wikipedia articles.

Dictionary Construction

This is the centerpiece of our system, and includes all potential string to entity pairs.
Entities were collected as follows:
- include all KB entities
- augment all articles in Wikipedia dump
- build equivalence classes using redirects and canonicalization
- choose representative article, KB entity if possible

Route_102_(Virginia_pre-1933)	State_Route_102_(Virginia_1928)
State_Route_102_(Virginia_1928-1933)	State_Route_102_(Virginia_pre-1933)
State_Route_63_(Virginia_1933)	State_Route_63_(Virginia_1933-1946)
State_Route_63_(Virginia_1940)	State_Route_63_(Virginia_pre-1946)
State_Route_758_(Lee_County,_Virginia)	**Virginia_State_Route_758_(Lee_County)**

Strings were collected as follows, producing three dictionaries:
- titles of articles, either literally or after deleting (...)
- titles of disambiguation pages (EXACT)
- performing lower case normalization (LNRM)
- doing fuzzy match (FUZZ)

String to entity maps are weighted using occurrences of the string as anchor text of the a link to the entity in:
- w Wikipedia text
- W In the Web

```
0.997642        Hank_Williams                   W:936/938 c d t w:756/758
0.00117925      Your_Cheatin'_Heart             W:2/938
0.000589623     Hank_Williams_(Clickradio_CEO)  c w:1/758
0.000589623     Hank_Williams_(basketball)      c w:1/758
0               Hank_Williams,_Jr.              d
0               Hank_Williams_(disambiguation)  c
0               Hank_Williams_First_Nation      d
0               Hank_Williams_III               d
```

In addition we also explored Google rank for string in en.wikipedia.org (GOOG).

Supervised Disambiguation

We train a multiclass classifier for each string. Given a context for the string, it returns a score for each corresponding entity in the EXACT dictionary.
Training data:
- collect occurrences of string as anchor text of links from Wikipedia
- also include remapped entities

Training features are extracted from spans of text consisting of 100 tokens to the left and 100 to the right of a link
- the anchor text;
- the lemmas in the span;
- lemma for noun/verb/adjective in a 4 token window around the anchor text;
- lemma and word for noun/verb/adjective before and after the anchor text;
- word/lemma/POS bigram and trigrams around the anchor text.

We trained SVMs using these binary features with a linear kernel.

Entity Linking Components

We evaluated two components for the 2010 TAC-KBP entity linking task:
- *cascade of dictionaries* (`run1`) — an overall context-independent score for each entity, which defaults to LNRM (lower-cased normalized) in case of an EXCT (exact match) miss;
- *heuristic disambiguation* (`run2`) — a context-dependent score, obtained by querying the longest matching entity name (based on a combination of NER, coference, and matching Wikipedia titles) against the context-independent dictionary.

For reference, we also include the results of some of our other entity-linking components from the 2009 TAC-KBP:
- *dictionary based on Google results* (`goog`) — the "GOOG" flavor of the dictionary;
- *distantly supervised disambiguation* (`supervised`) — a multi-class classifier trained on Wikipedia sentences;
- *Wikipedia knowledge* (`knowledge`) — cosine-similarity and TF-IDF and link scores from Wikipedia;
- *NIL baseline* (`nil`) — a baseline that always returns NIL.

Heuristic Disambiguation

Simple heuristic based approach that attempts to disambiguate an entity mention by identifying other possible mentions of the same entity in the text. The intution is as follows:
- Some mentions are more ambiguous than others
  Example: acronyms such as "ABC"
- Other mentions of the same entity in the document are less ambiguous
  Example: "American Broadcasting Company"
- To keep things simple, we assume that the longest entity mention is least ambiguous

Find the set of possible mentions for a given entity mention
- Run NER on document text to find all entity names
- For all occurrences of the target entity string, find the longest NER chunk it was part of.
- Use coref to find all entity names that are coreferent with the target entity.
- Find all matches of Wikipedia titles the target mentions can refer to in document

Identify Wikipedia title based on longest matching entity string
- Find the longest string obtained by extending the current entity mention to an NER chunk, coreference resolution, and matching Wikipedia titles in the document
- Use this longest string to query dictionary to get matching titles $T_{longest}$
- Take original mention to query dictionary to get matching titles T_{orig}
- Take interection of matching titles to get $T = T_{longest} \cap T_{orig}$
- Select the top-ranked title from this resulting set T.

Knowledge Based Disambiguation

We can use overlaps of context with KB and Wikipedia article text:
- Generate a TF-IDF index for Wikipedia
- Use similarity scores returned from search engine (Lucene)

Constructing the query:
- The mention alone, without the surrounding context.
- The last occurrence of the mention in the text, with a span of 25 tokens to the left and to the right of the mention.
- The concatenation of all matching mentions, and their 25 token spans, in the text.
- A window of 1,000 tokens around the last mention in the text.

Preliminary results on a development set showed that using the concatenation of all occurrences of the mention and their 25 word contexts performed best, and this was used for test set queries. After issuing the query, we filtered the returned list of Wikipedia articles and similarity scores, retaining only those covered by the dictionary. The rationale here was to match the same set of articles under consideration by the other methods.

Entity Linking Results on Training Data

Entity linking results for the 2009 TAC-KBP evaluation set (news data).

	Micro			Macro		
	3904 queries	1675 KB	2229 NIL	560 entities	182 KB	378 NIL
`run1`	0.7485	0.6949	0.7887	0.6851	0.5535	0.7485
`run2`	0.8215	0.8078	0.8318	0.7303	0.6737	0.7575
`goog`	0.7789	0.6955	0.8416	0.7135	0.5495	0.7924
`supervised`	0.7705	0.7039	0.8205	0.6963	0.5723	0.7560
`knowledge`	0.7029	0.5272	0.8349	0.6768	0.3765	0.8214
`nil`	0.5710	0.0000	1.0000	0.6750	0.0000	1.0000

Entity linking results for the 2010 TAC-KBP training set (web data).

	Micro			Macro		
	1500 queries	1074 KB	426 NIL	463 entities	462 KB	1 NIL
`run1`	0.8727	0.8799	0.8545	0.8174	0.8173	0.8545
`run2`	0.8507	0.8939	0.7418	0.8299	0.8301	0.7418
`goog`	0.8520	0.8873	0.7629	0.8244	0.8245	0.7629
`supervised`	0.8393	0.8808	0.7347	0.8163	0.8165	0.7347
`knowledge`	0.7740	0.7784	0.7629	0.7235	0.7234	0.7629
`nil`	0.2840	0.0000	1.0000	0.0022	0.0000	1.0000

Entity Linking Results on Evaluation Data

Entity linking results for the 2010 TAC-KBP evaluation set

	2250 queries	750 ORG	741 GPE	751 PER
`run1`	0.8000	0.7507	0.7183	0.9508
`run2`	0.7933	0.7813	0.6849	0.9134
`highest`	0.8680	0.8520	0.7957	0.9601
`median`	0.6836	0.6767	0.5975	0.8449

Entity Linking Conclusions

- We operate over all Wikipedia articles, returning NIL when an article not in the KB is selected
- Dictionary is the cornerstone of our system
- Dictionary includes counts, and performs very well: 80%
- Heuristic disambiguation using NER and coference worked well for news data but not web data, and did not give overall improvement over dictionary for 2010 evaluation set.
- Our system did not use free text from Wikipedia pages associated with the knowledge base nodes. We did not participate in the entity linking without Wikipedia pages task but our runs outperforms the systems that did participate.

Future Work - Combination

We plan to combine the heuristic disambiguation system with the robust system we developed for TAC-KBP 2009.
- *heuristic voting* — vote using inverse of rank of GOOG, the cascaded dictionary,
- *optimized mix* — a linear combination of positive weights applied to the scores from many components,