(Almost) Total Recall - SYDNEY_CMCRC at TAC 2012

Will Radford †‡ **Will Cannings** †‡ **Andrew Naoum** † **Joel Nothman** †‡
Glen Pink † **Daniel Tse** †‡ **James R. Curran** †‡

†ə-lab, School of Information Technologies
University of Sydney
NSW 2006, Australia

‡Capital Markets CRC
55 Harrington Street
NSW 2000, Australia

{wradford,anaoum,joel,gpin7031,dtse6695,james}@it.usyd.edu.au
me@willcannings.com

Abstract

We explore unsupervised and supervised whole-document approaches to English NEL with naïve and context clustering. Our best system uses unsupervised entity linking and naïve clustering and scores 66.5% B^3+ F1 score. Our KB clustering score is competitive with the top systems at 65.6%.

1 Introduction

Named Entity Linking is the task of grounding mentions of Named Entities (NEs) to knowledge base (KB) entries or, if the entity is not in the KB, NIL. The TAC KBP NEL tasks over the past two years have included the further task of clustering NIL mentions that refer to the same entity – a crucial task when building a KB.

In analyzing the errors made by our previous English NEL systems (Radford et al., 2010; Radford et al., 2011), we identified two key areas for improvement: candidate recall and supervised learning. NEL systems can often be analyzed as a process where NEs are *extracted* (trivial in the TAC case where the query name and offset are provided), *searched* for in the KB to generate a list of candidate entities and *disambiguated* to choose the best entity for each NE. Candidate generation is an important step in such a pipeline since if the correct entity is not retrieved, it cannot be chosen. Our first target was to boost recall to at least 95%, as reported in previous successful TAC systems (Lehmann et al., 2010).

Improving recall also results in a noisier candidate list, potentially making the disambiguation problem more difficult. Many of the top systems in TAC 11 used a supervised model for disambiguation (Ji et al., 2011), which has the advantage of combining evidence in a more principled manner. In addition to these main goals, we wanted to experiment with some more sophisticated NIL clustering methods.

2 Data Preprocessing

We link against the Wikipedia dump from April 2012[1]. Entity aliases are extracted from article titles, redirects and titles of disambiguation pages that link to the article and are indexed in Apache Solr 4[2]. The aliases are treated as untokenized strings normalized for case, diacritics and unicode character (Normalization Form Compatibility Composition). The article wiki markup is processed and the results stored in a variety of Tokyo Tyrant[3] key-value stores. These can be used to lookup the text, inlinks, outlinks and categories for a given title. We also calculate statistics over the graph of links between entities including: *entity prior* – the number of links to an article normalized by total number of articles – and *reference probabilities*, the conditional probability of linking to an entity given a particular alias (i.e. $p(\text{entity}|\text{alias})$). We predict and store the NE type of the entity based on several features of its article (Nothman et al., 2012).

2.1 Crosswikis Aliases

We use the Crosswikis dataset (Spitkovsky and Chang, 2012) to provide a wider set of entity aliases

[1] http://dumps.wikimedia.org/

[2] http://lucene.apache.org/solr/

[3] http://fallabs.com/tokyotyrant/

drawn from pages *outside* the Wikipedia article graph. These entity aliases, anchor text of incoming links, should be higher coverage but noisier. We apply a similar normalization process as the Wikipedia anchors: case, diacritic and unicode character. Multiple whitespace is replaced with single space characters, leading "`the `" and "`list of `" removed, and dipthongs (`oe` → `o`, `ue` → `u`) normalized. The target Wikipedia redirect URLs were resolved to their eventual article title. We excluded aliases that resolved to an empty string after normalization, appeared less than three times in the dataset, or had a reference probability less than 5%. The remaining aliases are added to the Solr index for their entity.

2.2 Generated Aliases

From the list of redirects for each entity in Wikipedia, we extracted common transformation rules from an entity name to its corresponding redirect. These rules could then be applied to other entities to generate additional aliases. We extracted the transformation rules using Levenshtein edit distance to determine common subsequences between string pairs, and then replaced common words with wildcards. For example, the entity "`Valve Corporation`" has the redirect "`Valve`". From this entity-redirect pair, we determine that common words are "`Valve`". Replacing "`Valve`" with the wildcard "`<1>`" results in the transformation rule "`<1> Corporation` → `<1>`". This transformation rule can be applied to any entity name that matches the left-hand side of the rule. For example, we applied the aforementioned rule to "`Oracle Corporation`" to generate the alias "`Oracle`".

From the 663,624 transformation rules that we automatically extracted, we manually curated a list of 434 rules that occurred most frequently. Rules included transformations such as the removal of name titles, prefixes, suffixes and middle initials; the abbreviation and removal of organisation suffixes; and the abbreviation and removal of state and country names. We then applied the curated list of rules to all applicable entities in our Wikipedia dataset. The resulting generated aliases are added to the Solr index for their entity.

3 Named Entity Linking

We take a whole document approach to Named Entity Linking despite the TAC task only focussing on one query per document. Although this means we perform more processing than strictly necessary, we believe that it better captures local document context, helping linking decisions.

3.1 Candidate Generation

We tokenize and extract Named Entities using the C&C Tools (Curran et al., 2007) with a 4-class (PER, ORG, LOC, MISC) model trained on approximately 1600 Australian newswire stories from 2009. We then need to resolve the query name to one of the extracted NEs, creating a dummy NE if no full or partial match could be found. The offsets provided with the TAC 12 queries make this process substantially easier.

3.1.1 In-document Coreference

We identify *chains* of NEs using simple coreference rules. NEs are sorted by length with longest first and each is processed in turn to find the best coreference match. The matching algorithm normalizes the NE for case and removes titles such as "Mrs". Exact matches to previous NEs are preferred (i.e. "`Ms Gillard`" or "`Gillard`" matches "`Gillard`"), then non-uppercase unigram suffix matches (i.e. "`Gillard`" matches "`Julia Gillard`"), then non-uppercase unigram prefix matches (i.e. "`Julia`" matches "`Julia Gillard`"), then acronym matches where the initial upper-case characters (excepting stopwords) of the NE (i.e. "`DoJ`" matches "`Department of Justice`"). Since we are coreferring NEs, these rules do not handle nominal or pronominal coreference. We noticed worse NER performance at the beginning of sentences where capitalized words were misidentified as NEs so we add aliases missing their initial token for any sentence-initial NEs.

3.1.2 Query Expansion

We apply rules to extract more in-document evidence when searching for candidate entities. We maintain a list of backoff queries to apply if there are no hits for the first query. We exclude any single word NE mentions that are substrings of the longest NE (i.e. "`Julia`" would be excluded if "`Julia

Gillard" is in the same chain) since we assume they are less specific. If the NE had been resolved to the query name, the name is added to a backoff list since there is not always perfect correspondence between names and NEs.

State aliases to the right of NEs are expanded and added to the query (i.e. "Austin, TX" will add "Austin, Texas"). Organizational suffixes such as "Inc" and "GmbH" are removed and the resulting NE added to the query. Bureaucracy poses difficulties for NEL since some organizations have common and particularly ambiguous names. It is feasible, for example, for any country to generate an entity "Department of Foreign Affairs, <Country>". If country names are found in the document and any NEs start with "Ministry", "Department" or "Office", a query of the NE and these country names are searched *first* and the other queries used as backoff.

3.1.3 Search

The expanded query is used to search the index using the Wikipedia, Crosswikis and generated aliases. The top 100 results are boosted by their *entity prior* (inlink count) with title and redirect matches weighted (weight = 100) more than disambiguation redirects, crosswiki and generated aliases (weight = 10). At this point we are able to measure recall, defined as the proportion of queries for which the candidate entities contains the correct entity from the gold-standard. For a KB query to count to the *recallable* total, the gold entity can be at any rank and NIL queries count to the total, regardless of their candidate entities. Recall for specific systems on datasets are reported in Table 2.

3.2 Unsupervised Linking

Once candidate entities have been retrieved for each coreference chain, we use a sequence of processing components to extract features for each candidate, then combine them for a final score by which entity candidates can be ranked. The features are described below.

Entity Match *Reference probability* is calculated for the entity and the longest NE in the chain. The *entity prior* is also used. *Entity title dice similarity* is calculated between the character bigrams from entity title and longest NE in the chain.

Document Context *Category score* and *Context score* specify an entity's relation to the whole document (Cucerzan, 2007). The former is the overlap of the entity categories with those of other candidates of other chains in the document, with a penalty to correct for the entity's categories. "Contexts" are extracted from the entity article: the anchor text of reciprocal links (i.e. $A \leftrightarrows B$) and those from the first paragraph. An entity's *context score* is summed frequency of all its contexts in the document. Both scores are normalized by the total of their respective values over all candidates for all chains in the document.

Graph Context The entity candidates are first ranked by the sum of their *category score* and *context score*. Then, assuming that the ranking is at least reasonable, the top ranked entity for each chain is examined and the entities that link to them added to a *document inlink set*. In a second pass through the chains, each candidate entity is assigned a *inlink overlap* score. This is the log of the size of the intersection between the entities that link to this candidate and the document inlink set.

Title Context This is a measure of compatibility with titles of other entity candidates in the document. In the sentence "The team toured Ontario, starting in Melbourne.", "Melbourne" refers to Melbourne, Ontario rather than the more prominent Australian city Melbourne. If the entity Ontario is a candidate for another coreference chain in the document, it should reinforce Melbourne, Ontario as a candidate. First, we extract context from each candidate of each coreference chain to try to identify *context-bearing entities* (eg. "Ontario" in Melbourne, Ontario). Context here refers to non-parenthesized tokens after a comma in the candidate title. Then, we check to see if that context matches the title of any other candidate to identify *supporting entities* (eg. Ontario). Each entity's supporting entities can be sorted by the distance (number of sentences) from the entity. Each *context-bearing entity* is scored 1 if there is a supporting entity with an extra bonus point for being the closest and a further point for being in the same sentence. As such, `Melbourne, Ontario` would be scored 3 since it is the closest match in the same sentence supported by the candidate `Ontario` for the chain containing "Ontario".

The final score is the average of the following features: *Reference probability*, *Entity prior*, *Category score*, *Context score*, *Inlink overlap* and *Title context*.

3.3 Supervised Linking

Supervised systems incorporate many sources of evidence to inform the linking decision in a principled way. Features can examine the query string, the query document, as well as the candidate entity and its article.

To facilitate supervised linking, labelled training data (gold-standard concepts) must be available. In our experiments, we use query data from past TAC years as training data: these come with gold standard entity IDs, and the data format is comparable with TAC 12 for easy integration.

We drew the design of our supervised features from two TAC 11 systems – (Anastácio et al., 2011) (including LDA features) and (Zhao et al., 2011) (Wikipedia link structure features), which are representative of features typically used in supervised entity linking.

Wikipedia Link Structure

- *Reference probability, Entity prior.* As above.

Entity Title-Chain Similarity

- *Alias cosine similarity.* The maximum character bigram cosine similarity between the mentions in the coreference chain and all of the candidate aliases for an entity.

- *Entity title dice similarity, Entity title cosine similarity.* As above with a variant that uses cosine distance.

- *Entity title begins/ends with query, query begins/ends with entity title.* Whether the entity article title begins/ends with a substring of the query name, or vice versa.

- *Entity title is substring of query, query is substring of entity title.* Whether the entity article title subsumes the query name, or vice versa.

- *Entity title edit distance/Jaro-Winkler distance.* Levenshtein distance or Jaro-Winkler distance computed between the entity article title and query name.

- *Article-query document cosine similarity.* Cosine similarity between the term vectors of the entity article and the query document.

- *Acronym match.* Whether the query is an acronym of the entity title.

Entity Type

- *Entity type matches.* Whether NER on the query string yields the same NE type as the candidate Wikipedia page's predicted NE type.

- *Mention preceded by locative P and is location.* Whether the query string is of type LOC, and is preceded by a locative preposition.

Topic Modelling

We trained an LDA model using the Vowpal Wabbit online machine learning toolkit, [4] with training parameters $k = 100$ (the number of topics), $\alpha = 1$, $\rho = 0.1$, on documents from TAC 09 queries and the Wikipedia articles from April 2012.

Similarity according to the topic model is integrated into supervised linking as a feature:

- *Topic similarity.* The Hellinger distance between the predicted topic distribution of the query document and entity article, both using stemmed tokens.

Feature weights are acquired with the maximum entropy learner MegaM [5] using the `binomial` mode.

The supervised linking model used for our TAC 12 run employs a reranking step which uses the output of an unsupervised linker to eliminate poor candidates on the basis of features which are cheap to compute. Candidates whose unsupervised linking score falls below a threshold are pruned and are not considered by the supervised model. This has two benefits – the reduced candidate set is less noisy, and the supervised features which are more costly to compute can operate over a focused set of candidates.

[4] `http://hunch.net/~vw`

[5] `http://www.cs.utah.edu/~hal/megam`

ID: Linking / Clustering	All	KB	NIL	NW	WB	PER	ORG	GPE
Highest	73.0	68.7	84.7	78.2	64.6	84.0	71.7	69.4
1: Unsupervised / Naïve	66.5	65.6	67.5	70.0	59.8	73.9	59.8	63.1
2: Supervised / Naïve	61.0	56.8	65.6	64.6	53.8	71.2	55.6	51.4
3: Unsupervised / Context	58.8	65.6	49.1	61.3	53.7	60.0	53.0	62.3
4: Supervised / Context	54.0	56.8	48.9	57.0	47.9	59.0	49.3	50.7
Median	53.6	49.6	59.4	57.4	49.2	64.6	48.6	44.7

Table 1: B^3+ F1 scores over TAC 12 data

3.4 Clustering

We use two clustering methods: *naïve* , based on (Radford et al., 2011), and *context*. These are applied after unsupervised or supervised linking to the resulting queries and their ranked candidates. Naïve clustering takes advantage of the fact that we link against a larger and newer version of Wikipedia than the TAC KB. Queries linked to KB nodes are assigned an entity ID if the Wikipedia title maps to a KB node otherwise a NIL ID is constructed for that entity. If the query was linked to NIL (i.e. we could not find *any* candidates since we do not use a threshold or NIL classifier), we assign a NIL entity ID based on the query name. This approach was used for CMCRC 3, our best run for TAC 11 (Radford et al., 2011).

In effort to move beyond naïve baselines, our other clustering method uses the *context* of each query's coreference chains. The clustering is implemented using the hierarchical clustering package from SciPy[6] using `single` linkage method and `cosine` metric. Queries linked to both KB and NIL entities are clustered with the following features: untokenized query name, unigram counts from sentences containing NEs from the query's coreference chain. The latter are lower-cased and filtered to remove stopwords. Clusters are flattened using the `distance` threshold of 0.5 and if it contains a Wikipedia title mappable to the TAC KB, that is chosen as the final ID, otherwise a NIL ID is generated. For example, if two queries named "Tom Cruise" cluster together and the first had been linked to the TAC KB entry for `Tom Cruise` and the second to NIL, both will inherit the appropriate entity ID. Entity ID disagreements are resolved by choosing one at random.

System	Data	R	All	KB	NIL
Unsupervised	11	96	86.6	83.5	89.7
Supervised	11	94	84.7	78.6	90.8
Unsupervised	10	97	85.2	82.5	87.4
Unsupervised	09	95	81.5	76.9	85.4
TAC 10	11	n/a	77.9	67.4	88.4
TAC 10	10	n/a	84.4	79.0	88.8
TAC 10	09	n/a	77.7	72.6	81.6

Table 2: Linking accuracy over previous TAC evaluation datasets. Performance for the TAC 10 system (CMCRC 1) is as reported in previous papers and is the same linking system used in TAC 11. Note that we only report supervised numbers trained on TAC 09 and tested on TAC 11. Recall percentage (R) is reported where available.

4 Results

The structure of our system allows us to combine a linking and clustering system. As such, our four runs are the different combinations of our supervised and unsupervised linkers and naïve and context clusterers. Table 1 shows the results of our systems on the TAC 12 dataset with top and median listed for comparison. Our best system combines an unsupervised linker and naïve clusterer and, at 66.5% B^3+ F1, performs well relative to the top and median scores. However, our KB B^3 F1 score at 65.6% is more competitive, only 3.1% from the top score, reflecting the higher priority we place on improving linking over clustering. Our NIL context clustering scores are substantially below median where the naïve clustering scores are above, suggesting these coarser methods (without distance parameters, etc.) are more robust to any differences between the TAC 11 and TAC 12 datasets.

Table 2 shows the linking performance of different versions of our system over different datasets.

[6] `http://scipy.org`

ID: Linking / Clustering	All	KB	NIL
1: Unsupervised / Naïve	84.2	82.5	86.3
2: Supervised / Naïve	82.1	77.3	87.5
3: Unsupervised / Context	84.3	82.5	86.5
4: Supervised / Context	82.3	77.3	87.9
TAC 11 (CMCRC 3)	75.4	n/a	n/a

Table 3: B^3+ F1 score over TAC 11 evaluation data. The TAC 11 system draws NIL ids from Wikipedia pages not in the KB.

The top section shows updated systems while results from the bottom section are drawn from previous system reports (Radford et al., 2010; Radford et al., 2011). Our work this year concentrated on improving performance on the TAC 11 dataset and we achieved an 8.7% increase in linking accuracy from our TAC 10 system that we also used for TAC 11. The gains on previous datasets are more modest, almost certainly because we optimized for gold-standard links and NIL clusters in the TAC 11 data. We also report recall (as defined in Section 3.1.3) for the new systems and have found that this is a valuable tool for analysis and debugging linker errors.

We remain frustrated by our supervised system's poor performance relative to our unsupervised system where experience and previous results suggest that the reverse should be true (Ji et al., 2011). While one clue is the lower recall than for the unsupervised system, we have found supervised NEL a complex problem to analyze and solve. This is in part due to the instability of instance generation (a different search strategy can change the instances for learning and classification) and challenges from modelling the NIL link.

Table 3 shows the B^3 clustering scores of our TAC 12 systems on the TAC 11 data. Again, unsupervised tends to perform better than supervised, but context clustering is more successful than naïve. This is surprising given our results in Table 1 showing that naïve clustering performs better on TAC 12 data.

5 Conclusion

Our systems in TAC 12 explore unsupervised and supervised whole-document approaches to NEL with naïve and context clustering. Our best system uses unsupervised entity linking and naïve clustering and scores 66.5% B^3+ F1 score. Our KB clustering score is competitive with the top systems at 65.6%.

References

[Anastácio et al.2011] I. Anastácio, I.I.D. Lisboa, B. Martins, and P. Calado. 2011. Supervised learning for linking named entities to knowledge base entries. In *Proceedings of TAC 2011*.

[Cucerzan2007] 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.

[Curran et al.2007] 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.

[Ji et al.2011] Heng Ji, Ralph Grishman, and Hoa Trang Dang. 2011. Overview of the TAC2011 knowledge base population track. In *Proceedings of TAC 2011*.

[Lehmann et al.2010] John Lehmann, Sean Monahan, Luke Nezda, Arnold Jung, and Ying Shi. 2010. LCC approaches to knowledge base population at TAC 2010. In *Proceedings of TAC 2010*.

[Nothman et al.2012] Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R. Curran. 2012. Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence*. (in press).

[Radford et al.2010] Will Radford, Ben Hachey, Joel Nothman, Matthew Honnibal, and James R. Curran. 2010. Document-level entity linking: CMCRC at TAC 2010. In *Proceedings of TAC 2010*.

[Radford et al.2011] Will Radford, Ben Hachey, Matthew Honnibal, Joel Nothman, and James R. Curran. 2011. Naive but effective NIL clustering baselines – CMCRC at TAC 2011. In *Proceedings of TAC 2011*.

[Spitkovsky and Chang2012] Valentin I. Spitkovsky and Angel X. Chang. 2012. A cross-lingual dictionary for English Wikipedia concepts. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, Istanbul, Turkey, May.

[Zhao et al.2011] Y. Zhao, W. He, Z. Liu, and M. Sun. 2011. Thunlp at tac kbp 2011 in entity linking. In *Proceedings of TAC 2011*.