Benchmarks for Enterprise Linking: Thomson Reuters R&D at TAC 2013

Tom Vacek[†] **Hiroko Bretz**[†] **Frank Schilder**[†] **Ben Hachey**[‡]

[†]Thomson Reuters R&D
610 Opperman Drive
Eagan, MN 55123, USA

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

{thomas.vacek,hiroko.bretz,frank.schilder}@thomsonreuters.com
ben.hachey@sydney.edu.au

Abstract

This paper describes the TRRD systems entered in the TAC 2013 entity linking challenge. We explore a restricted version of the task that accesses only an entity authority file with (possibly noisy) alternative names and plain text from the target domain. This is designed to reflect the problem of linking to existing entity authorities within companies like Thomson Reuters. We used the 2013 shared task to benchmark this task setting.

1 Introduction

Entity linking provides a framework for authority-driven named entity linking and resolution (Ji and Grishman, 2011; Hachey et al., 2013). This is an exciting development for organizations that have existing, curated entity authorities. The use of Wikipedia as a wide-coverage knowledge base has driven recent work, but, as a result, state-of-the-art systems rely on rich category, infobox and link information (Bunescu and Paşca, 2006; Cucerzan, 2007). This information is not typically available in enterprise authorities, where a multitude of entities are often not covered by Wikipedia yet, or are excluded due to Wikipedia's notability guideline.

The original TAC linking formulation shares the motivation of using Wikipedia as a means to bootstrap a wider-coverage KB (NIST, 2009). However, general practice of task participants is now to exploit all available information in Wikipedia, including use of a more recent version. This represents a setting that uses a current KB to link historical data. It also is makes it trivial to determine NIL status and cluster links that have no corresponding node in the smaller TAC KB. This is a logical consequence of the competition's resources, but makes it difficult to generalise findings to our setting.

We explore a restricted version of the KBP entity linking task that accesses only an entity authority file with alternative names and unannotated text from the target domain. Specifically, we create an authority from the distributed KB that contains only the entity full name and aliases. Aliases are obtained by stripping appositions from titles and by collecting anchor text from links in the KB fact elements.

We use the 2013 shared task to benchmark this restricted task setting against systems that use richer information (recent Wikipedia, KB `wiki_text`, KB `fact` text, etc). We also present results for systems that use alias weighting and text from the TAC KB for comparison. Our baseline for the restricted task performs near 80% of the top score. This is encouraging given the difference in KB resources, but the gap is substantial. In current work, we are exploring ways to address this gap by bootstrapping Wikipedia-like structure from massive historical text collections.

2 Approach

Our core pipeline comprises: a lookup tagger that performs simultaneous authority-driven mention detection and candidate generation; a disambiguator that performs simultaneous ranking and NIL detection using a binary SVM classifier. The approach here is based on an existing company tagging and resolution system (Thomas et al., 2014). We adapt it here for query-driven linking.

2.1 Lookup tagger

The first step in the linking pipeline is KB-driven candidate generation. Given a query mention, this finds resolution candidates through approximate matching against names from the authority. It aims for high recall, leaving candidate ranking and NIL detection to the disambiguator (Section 2.2 below).

The lookup tagger scans the query tokens, checking a dictionary that maps authority name tokens to KB identifiers. The candidate set is reduced to the intersection after each lookup. Token matching is not case sensitive. This recall-oriented approach leads to large candidate sets for common names (e.g., Smith). Conversely, many queries have no candidates due to the limited number of alternative names in the authority. Current work focuses on improving the recall of the lookup tagger, e.g., by generating abbreviations and other alternative names.

2.2 Disambiguator

The next step in the pipeline is disambiguation. The key component here is a binary SVM classifier that is run for each candidate to determine whether it is a good resolution or not. If multiple candidates are classified as correct, then the one with the highest true score is returned. If no candidates are classified as correct, then it is not considered to be a mention of a known entity. For the purposes of TAC queries, NIL is returned.

We use a linear Support Vector Machine (SVM) classifier, as implemented in LIBLINEAR (Fan et al., 2008). While SVM is one of many possible choices for binary classification, the method is particularly attractive in this case for having theoretical error bounds that are independent of the dimension of the input (Vapnik, 1999), for having a way to control the tradeoff between precision and recall by unequal misclassification costs, and for giving confidence scores based on the normal distance from a test point to the decision boundary. All of these qualities are useful for the application. Unequal misclassification costs empirically improved system performance. Of course, the ranking aspect of the task requires confidence scores.[1]

[1]While there are a number of proposed methods to estimate the class conditional probability $\Pr(y|x)$ for SVM such as (Wu et al., 2003), we simply use the normal distance.

Table 1 contains the list of features used. These are divided into baseline and context features. The baseline features quantify how well the mention string matches an authority name. In addition, the baseline includes features to characterize the distribution of string matching statistics over the candidate pool (e.g., candidateSD, isOutlier). Context features attempt to quantify evidence for the candidate in terms of the semantic context. For instance, the existence of the string "flight" in proximity to "United" is strong evidence that the latter string refers to the well-known airline. The context model is derived from the `wiki_text` and `fact` elements of the distributed TAC KB.

2.3 KBP output formatter

Finally, the output is written to KBP format. This identifies the ID or NIL value from the disambiguator that corresponds to the query mention. Offsets are used if available, otherwise the last matching mention in the document is used. No clustering of the NIL entities is carried out and every NIL occurrence receives a unique number.

3 Configuration

3.1 Authority

Offline, we generate an authority from the distributed KB. This contains a long name (from the `name` attribute on `entity` elements). It also includes aliases: titles stripped of parenthesized expressions (e.g., `Ken Thomson (footballer)` $\mapsto$ `Ken Thomson`); titles stripped of appositions indicated by commas (e.g., `Saint Paul, Minnesota` $\mapsto$ `Saint Paul`); and anchor text from links inside the `fact` elements of the distributed KB. Names are marked as unambiguous if they do not occur in WordNet and are not homographic within the authority.

3.2 Train/test data

We generate training data for the disambiguation classifier from the KBP 2012 gold data. For each query document, we run the lookup tagger. Then we create instances based on query mentions. If there is a gold KB ID, this is used as a true instance and all others are false. If the gold annotation is NIL, then all candidates are used as false examples.

cosine	Cosine/TFIDF sim between mention and name list match. IDF weights were pre-extracted from an internal corpus of company names. Given as value and as indicators for logarithmic bins.
candidateSD	Binary: Is candidate's cosine score close to the max of all the candidates?
isOutlier	Binary: Is candidate's cosine score a high outlier among all the candidates for the mention?
isUniqueOutlier	Binary: Does the candidate have the highest cosine score and no other candidate is within one standard deviation?
highestCosine	Highest cosine in candidate poo
levenshtein	Normalized Levenshtein (edit) distance between mention string and name list match.
isOneWord	Binary: Is the mention string one word?
candidateLength	Log character length of name list match.
numTokens	Number of tokens in the candidate.
candidateNum	Number of candidates, given as log value and indicators for log bins.
anchorCount	Count of the number of times the candidate is used as an anchor text in the authority, as log
isLongName	Binary: Indicates whether the lookup tagger found the candidate by its full name in the authority, not a synonym.
isLongNameMatch	Binary: Similar to isLongName, but with the further restriction that the mention string is an exact match.
longNameMatchInDoc	Binary: Indicates whether the candidate's long name was matched exactly anywhere in the document.
unambiguousAuthID	Binary: The document contains an unambiguous synonym (based on the unambiguous names list) for the candidate.
otherWordInDoc	Binary: Indicates whether the document contains another word from candidate's full name in the authority.
	context features
authorityMass	Fraction of words in wiki text and facts that are in common with document text.
contextMass	Fraction of words in document text that are in common with wiki text and facts.
factCount	Log of number of unique facts that have words in common with document.

Table 1: List of features for classifying candidate entities. Submission TRRD1 uses the first set of features. Submission TRRD2 uses all.

Data	System	Acc	$P_{\in}$	$R_{\in}$	$P_{\notin}$	$R_{\notin}$
2011	TRRD1	68.8	66.3	48.5	70.2	89.0
2011	TRRD2	70.1	69.2	51.3	70.6	88.9
2013	TRRD1	62.8	59.3	45.0	65.0	80.5
2013	TRRD2	62.2	59.1	47.1	64.2	77.2

Table 2: Accuracy (Acc), KB presicion ($P_{\in}$), KB recall/accuracy ($R_{\in}$), NIL precision ($P_{\notin}$) and NIL recall/accuracy ($R_{\notin}$).

3.3 Training and parameter settings

We used LIBLINEAR for the classifier, training on previous years' competition gold data. Model parameters were chosen using 2012 competition gold data for training and 2011 for testing. We found the best performance by putting a low misclassification cost on the negatives, but a 10-times larger cost for the positives. We used LIBLINEAR's L2-regularized, L2-penalized SVM primal solver.

4 Results

4.1 Runs

We submitted two runs to the 2013 evaluation. Neither access the web during the evaluation. Both use the offsets in the query to identify the entity mention within the document text. Neither generate meaningful confidence values. For each candidate, we produce a single candidate with 1.0 confidence or NIL. The runs differ in their use of a context model (derived from `wiki_text` and `fact` elements in the distributed KB).

TRRD1 The first submission is our baseline. It does not use the context features in Table 1.

TRRD2 The second submission also includes the `context` features from Table 1.

4.2 Official results (initial)

Table 2 contains development results on 2011 data (training 2012) and initial 2013 official results (training 2012). Note that these results and the corresponding analysis are based on gold data distributed prior to the workshop. For updated results on post-workshop gold distribution, see Section 4.5. The precision and recall numbers are calculated like accuracy. They evaluate the non-clustering portion of the task that simply returns NIL if no link is found.

System	2011	2013
TRRD1	79%	78%
TRRD2	81%	77%

Table 3: Accuracy of restricted configurations as percentage of top scores (2011: 86.8, 2013: 81.0).

Overall accuracy on the 2013 evaluation data is much lower than on the 2011 development set. However, the top system scores – 86.8 and 81.0 respectively for 2011 and 2013 – vary almost as much. In fact, our restricted task baselines perform consistently near 80% of the top reported score (Table 3). High NIL recall reflects a conservative approach to linking. This can be appropriate, e.g., when used as an assistive technology for human curators.

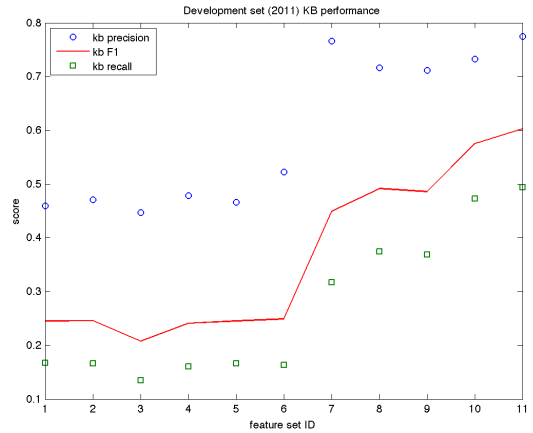
4.3 Feature analysis

1	cosine only
2	+ levenshtein
3	+ candidateNum
4	+ candidateLength, numTokens, isOneWord
5	+ candidateSD, highestCosineScore, isOutlier, isUniqueOutlier
6	+ isLongName, isLongNameMatch
7	+ longNameMatchInDoc
8	+ otherWordInDoc
9	+ unambiguousAuthID
10	+ anchorCount (TRRD1)
11	+ context features (TRRD2)

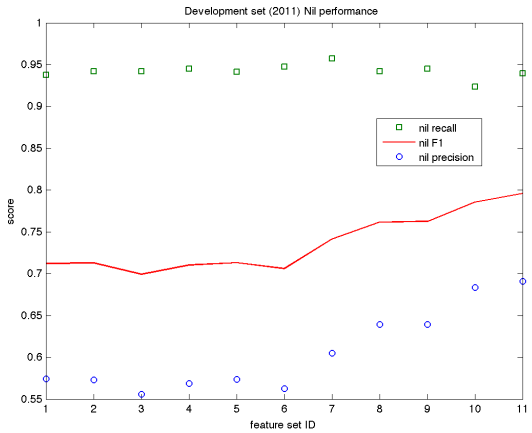
Table 4: Feature groups for additive analysis.

The features were engineered in groups, with each group intended to summarize some underlying property of the data. We provide an additive analysis of how each group contributes to the overall system. Figure 1 shows the results for the 2011 and the 2013 test sets while we trained for both on the data from 2012. The feature groups are described in Table 4.

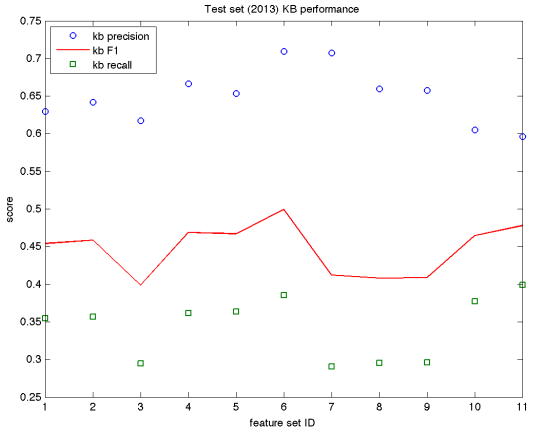
The additive study for 2011 test results shows that feature groups 7, 8, 10, and 11 have a significant impact on KB precision. Those features are based on the occurrence of long names or single words in the rest of the document and the number of links to the anchor in the knowledge base. Groups 7 and 8 are



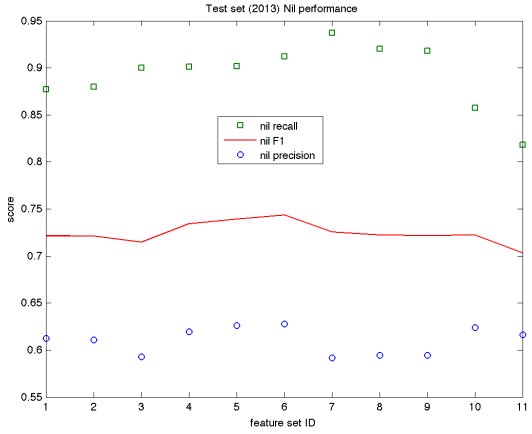
(a) kb



(b) nil



(c) kb



(d) nil

Figure 1: 2011/2013 Additive feature analysis. The x-axis corresponds to the feature groups in Table 4. Note that the less sophisticated feature groups (1-6) performed better in 2013 than on the 2011 validation set, whereas the the more advanced feature groups (7-11) failed to generalize in 2013.

useful for news articles because journalistic guidelines require that names are introduced with a full form and other variations of the same name are also often used in the article in order to avoid repetitions. The anchor feature is useful for matching to the most likely match and ignoring less frequent candidates.

We hypothesize that the increase in forum data led to the lower performance in 2013 because such text will not follow the journalistic guidelines that make the above mentioned features such strong features for the 2011 data.

There was a bug in the feature generation code that caused a serial number assigned to each query to be included in the features used to generate our submitted results. The feature analysis uses the corrected configuration (no serial numbers), whereas Table 2 retains the bug in order to be consistent with the submitted system.[2] The bug caused lower performance on the development set as would be expected, but gave improved performance on the 2013 queries, in particular a roughly 7% gain in KB recall. Remarkably, the addition of this dummy feature changed the weights of the learned model in a nontrivial way; a number of weights changed sign, and several changed significantly in magnitude. The weight given to the query serial number was positive and would make the classifier slightly more likely to make a link for a query with a higher serial number, other things equal. We attempted to understand what properties about the queries would have made this feature generalizable, but we were not able to find anything conclusive. Nevertheless, it may be worth randomizing future test sets to ordering effects.

In the additive analysis, we were surprised to see that feature group 6, which uses only features related to the tokens in the query, performed the best for the 2013 evaluation. The remaining features, which gave significant performance improvements on the 2012 task, actually hurt performance. Therefore, we set about to see if we could find a subset of features optimal for the 2013 task. In short, the answer was that we could not improve upon feature group 6. In particular, this suggests that the context information and the coreference information, as we have designed it, did not help for the 2013 task.

[2]This accounts for the discrepency between the reported results in Table 2 and the precision and recall scores plotted in Figure 1

4.4 Error Analysis

We reviewed a sample of 100 queries from 2013. These include 41 errors for TRRD1 and 43 for TRRD2. For four queries, one configuration returns a KB node when the gold answer is NIL. TRRD2 commits this error three times, suggesting the context features make the model slightly less conservative. There are also two queries where both configurations are incorrect. In both, the gold answer is a KB node, but TRRD2 returns NIL while TRRD1 returns an incorrect candidate.

For TRRD1, 76% of errors are on queries that should have been resolved to a KB node. Of these, the vast majority (97%) are lookup errors (i.e., the lookup tagger did not have this variation). This suggests that obtaining reliable alternative names is the primary challenge in our restricted task setting. 60% of lookup errors are politically biased person nicknames from forums (e.g., Nobama, McLame). The remainder are ambiguous abbreviations (23%, e.g., J.B., CCF), spelling errors (10%, e.g., Michicgan, Kay Baily Hutchison), substrings (3%, e.g., Karl), and transliterations (3%, e.g., Guenter Verheugen).

24% of errors are on queries that should have been returned as NIL. Of these, 80% were found on wikipedia.org as of November 11th, 2013. These are cases where use of a more recent and more complete Wikipedia dump gives a distinct advantage. This allows resolving to a known Wikipedia article (e.g., Virginia Tech Hokies football, World Anti-Doping Agency), then returning NIL when there is no corresponding node in the TAC KB.

TRRD2 has more NIL errors at 30% (24% for TRRD1). TRRD2 also has fewer NILs found in the current wikipedia.org at 69% (80% for TRRD1). The breakdown of lookup errors is the same.

4.5 Official results (updated)

After the workshop, LDC released updated gold annotation that corrects some systematic GPE errors. This resulted in a small changes of approximately 0.5 points accuracy for both our systems. Interestingly, TRRD1 accuracy decreased while TRRD2 increased. This suggests at least that the context features are not detrimental on 2013 data as the original results suggested. The updated data had a larger effect on the top accuracy score, which is now 83.3.

5 Discussion

We presented benchmark systems for a restricted version of the official TAC linking task. This is designed to reflect the problem of linking to existing entity authorities within companies like Thomson Reuters. The most restricted version accesses neither Wikipedia nor the text content of the distributed TAC KB. Our optimal development feature combination did not generalize well to the 2013 test set, likely due to the addition of forum data. Nevertheless, the accuracy of the benchmark configurations as a percentage of the top-reported systems is stable across 2011 and 2013 at nearly 80%.

Our primary interest is in linking, not NIL clustering. We envisage enterprise applications of linking that can tolerate noise or be tuned for precision. However, for maintenance of high-quality authorities, we believe that linking and especially clustering are best used as assistive technologies to facilitate human curation. Therefore, we did not attempt a meaningful clustering solution and we focus on linking-oriented evaluation measures.

Error analysis demonstrates that obtaining reliable alternative names is the primary challenge in our restricted task setting. In current work, we are investigating techniques to derive information about entities from large amounts of text. We believe that this approach is more realistic for situations where a rich KB like Wikipedia is not available for the targeted universe of entities. More concretely, we are mining name occurrences from text by computing frequencies and co-occurrences with other terms or relationships in general. In addition to authority maintenance, we are interested in the adaptability of linking for enriching linked data ecosystems within and across specific professional domains.

References

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

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.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: a library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. 2013. Evaluating entity linking with Wikipedia. *Artificial Intelligence*, 194(0):130–150.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1158, Portland, Oregon.

NIST. 2009. Task description for knowledge-base population at TAC 2009. Accessed 22 June 2010 from `http://apl.jhu.edu/~paulmac/kbp/090601-KBPTaskGuidelines.pdf`.

Merine Thomas, Hiroko Bretz, Thomas Vacek, Ben Hachey, Sudhanshu Singh, and Frank Schilder. 2014. Newton: Building an authority-driven company tagging and resolution system. In Emma Tonkin and Stephanie Taylor, editors, *Working With Text: Tools, techniques and approaches for text mining*. Chandos Publishing Ltd, Oxford, UK. In press.

Vladimir N. Vapnik. 1999. *The nature of statistical learning theory*. Springer, 2nd edition.

Ting-fan Wu, Chih-Jen Lin, and Ruby C. Weng. 2003. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005.