Context-Based Entity Linking – University of Amsterdam at TAC 2012

David Graus, Tom Kenter, Marc Bron, Edgar Meij, Maarten de Rijke
ISLA, University of Amsterdam
Science Park 904, 1098 XH Amsterdam, The Netherlands
{d.p.graus,tom.kenter,m.m.bron,e.j.meij,derijke}@uva.nl

Abstract

This paper describes our approach to the 2012 Text Analysis Conference (TAC) Knowledge Base Population (KBP) entity linking track. For this task, we turn to a state-of-the-art system for entity linking in microblog posts. Compared to the little context microblog posts provide, the documents in the TAC KBP track provide context of greater length and of a less noisy nature. In this paper, we adapt the entity linking system for microblog posts to the KBP task by extending it with approaches that explicitly rely on the query's context. We show that incorporating novel features that leverage the context on the entity-level can lead to improved performance in the TAC KBP task.

1 Introduction

Entity linking is the task of linking mentions of entities in a document to corresponding entities in a knowledge base. Wikipedia is the prototypical knowledge base for this task: its articles cover a large number of concepts, and its inter-article link structure provides rich semantic information.

The TAC KBP entity linking track provides us with entity mentions occurring in context documents and a background knowledge base in which these entities might occur. The TAC KBP knowledge base is derived from Wikipedia and contains 818.741 entities of three types: persons (PER), organizations (ORG) and geo-political entities (GPE).

The entity linking task is commonly split in three subtasks: for each mention (i) retrieve disambiguation candidates from the knowledge base, (ii) find the right candidate or decide that the entity mentioned should be considered absent from the knowledge base. Finally, (iii) cluster all mentions of entities that are not present in the knowledge base.

In our submission to the TAC KBP entity linking track we apply a state-of-the-art entity linking system for microblog posts (Meij et al., 2012) and adapt it to the task of entity linking in the TAC KBP context. This system is designed to link entity mentions in microblog posts (documents of at most 140 characters which are noisy and full of shorthand and ungrammatical text) to relevant Wikipedia articles.

It does so in two separate steps. The first step is recall-oriented and analogous to candidate generation in the TAC KBP task. It consists of retrieving a (ranked) list of candidates for n-grams extracted from the source document. The second step consists of determining which of these n-grams are relevant to the source text and should be linked. A machine-learning approach is applied with high dimensional feature vectors as input. The features include textual properties of the n-gram in the microblog post, of its associated concept, the combination of the two, and features that involve the microblog post in its entirety.

For the TAC KBP task some adaptations to this approach are needed. In the entity linking track, we are provided with the n-gram to link for each query, so there is no need to predetermine which n-grams are relevant for linking. Furthermore, in the TAC KBP task we are provided with reference documents of greater length and of a less noisy nature, which may contain valuable contextual information.

In this paper we take both the recall phase of can-

didate generation and the precision phase of feature-based linking as a starting point. We extend it by incorporating two approaches that benefit from the context document and knowledge base structure. The first approach entails searching for entities in the document that are related to the candidate entity, to leverage the context at the entity-level. Our second approach performs joint disambiguation in a document by considering all ambiguous entity mentions simultaneously. We combine and evaluate these approaches to determine to what extent we can leverage the context to improve entity linking performance.

Our main contribution is twofold: we provide a novel approach to entity linking by leveraging a query's context on the entity-level. Furthermore we compare two different approaches to context-aware entity linking and show that combining local clues with a context aware approach can increase entity linking performance.

2 Approach

We model the entity linking task as a classification problem. Our system layout follows a typical monolingual entity linking system architecture (Heng et al., 2011) and is divided in three subprocesses: (i) candidate generation, (ii) candidate disambiguation, (iii) NIL detection and clustering. No query expansion is performed.

In the candidate generation step, it is useful to employ a more recent knowledge base than the one provided by the TAC KBP task; entries found in the larger KB that can not be mapped to the TAC KB are likely to be NILs. Furthermore, a larger and more recent knowledge base provides more and more elaborate information. We use a Wikipedia dump of January 4 2012 as our background knowledge base. This particular Wikipedia snapshot contains 3.717.827 entities. For each query we generate a list of candidates by performing a search through this Wikipedia snapshot.

In the second step – the disambiguation – candidates are ranked in decreasing order of relevance. We apply supervised machine learning to several features derived from each of the candidate-query pairs that are generated in the first phase. A correct query-candidate pair is considered a positive example, an incorrect pair a negative example. The TAC KBP 2011 set was used to obtain ground truth for the training material. Based on the confidence score of the random forest classifier we select the highest scoring candidate for each query.

As a third and last step we cluster the queries for which no suitable candidate could be found together (the so-called NILs).

In the following sections we describe each subprocess in more detail.

2.1 Candidate generation

Candidate search is similar to the method of lexical matching for concept retrieval described in Meij et al. (2012). For each query, we use the entity mention as input for a search over Wikipedia article titles, disambiguation pages and anchors to return a set of disambiguation candidates. We find on average 336 candidates per query.

2.2 Disambiguation

Disambiguation is performed with three different feature sets and combinations thereof. In what follows we have candidate c, string s and query q occuring in reference document r.

2.2.1 Baseline features

Our baseline uses a subset of the features in Meij et al. (2012) further detailed in table 1. The features involve only a query and its candidate (no context is used) and include measures such as the overlap of the query with the title of the candidate, edit distance between the two, the number of inlinks and outlinks the candidate has and its commonness (Milne and Witten, 2008). Also the number of redirects and categories is taken into account as are features about the position and frequency of the query in the first sentence/paragraph of a candidate's text. We arrive at a total of 32 baseline features.

2.2.2 Context features

As the relatively noise-free and clean context documents of the queries can contain valuable information, we extend this baseline model with features that aim to use this context for disambiguation.

Table 1: Baseline features

Feature	Description
REDIRECT	Number of redirect pages linking to c
CAT	Number of categories associated with c
WLEN	Number of terms in title of c
CLEN	Number of characters in title of c
INLINKS	Number of pages linking to c
OUTLINKS	Number of page c links to
GEN	Function of depth of c in knowledge base category hierarchy
TF-T	Frequency of q in normalized title of c
TF-S	Frequency of q in normalized first sentence of c
TF-P	Frequency of q in normalized first paragraph of c
POS	Position of the first occurence in the first paragraph
NCT	Does q contain the title of c
TEN	Does title of c equal q
TCN	Does title of c contain q
EDIT-DIST	Levenshtein Distance between query and candidate title
CMNS	Commonness. Chance of c being target of link with q as anchor
APPOS	Apposition. Does the clarifying term in the title of c (e.g. 'band' in 'Alabama (band)') occur in the source text

The context features make use of the semantic information encapsulated in the graph structure of the knowledge base.

We try to discover related entities, entities that link from or to the candidate, in the context document. We achieve this by searching for occurrences of their various surface forms. Next to titles, which offer a canonical description of the entity, we leverage anchor texts of these entities. They tend to be less canonical and give us a real world natural language representations of the entity.

The context features are described in more detail in table 2. They are based on several properties of the surface forms found: the amount of titles and anchors occurring in the reference document, common metrics associated with the anchors, such as link probability, commonness, sense probability and distance to the query, and finally on the entities that are represented by the surface forms.

We normalize all features to measure the proportion of anchors or entities matched as well as the plain (scaled) frequency. We arrive at a total of 40 context features.

2.2.3 LOD features

An issue with methods that rely on the set of entity strings (S) co-occurring with the query entity (q) for disambiguation is that these co-occurring entities are ambiguous themselves. For example, using an undirected graphical model and exact inference to obtain the most likely assignment of candidate disambiguations to all entity strings requires the evaluation of all $|C|^{|S|}$ possible assignments, assuming that every entity has $|C|$ disambiguation candidates.

A cheaper way of performing this type of joint disambiguation is to consider all disambiguation candidates of entities in S simultaneously (Cucerzan, 2007). We adopt this approach and model individual disambiguation candidates as vectors of objects ($v(c)$). The context document (D) of q is then represented by: $D = \sum_{s \in S} \sum_{c \in C_s} v(c)$,where C_s is the set of candidates for entity string s. We use the scalar product between the query candidate disambiguation and this document vector representation to find the most similar query disambiguation candidate.

A popular approach to determine the relatedness of disambiguation candidates is to use the Wikipedia

Table 2: Context features

Feature	Description
T-OVL	Title overlap. Does the title of c occur in r.
A-OVL	Anchor overlap. Number of anchor texts of c occuring in r.
E-OVL	Entity overlap. Number of entities related to c for which anchor texts match in r.
INLINKS	Number of inlinks a related entity has. Separate for inlink entities and outlink entities
OUTLINKS	Number of outlinks a related entity has. Separate for inlink entities and outlink entities)
PROP	How often is an anchor text used to refer to an entity relative to the total amount of anchors referring to that entity. Aggregated over all matching anchors
LINKPROB	How likely it is for a string s to be used as an anchor text. Aggregated over all matching anchors
CMNS	How many times is an anchor text s used to refer to an entity relative to the total amount of times it occurs as an anchor. Aggregated over all matching anchors
SENSEPROB	How likely it is for an anchor text s to be linking to an entity. Aggregated over all matching anchors
DISTANCE	The amount of words between the anchor and query mention. Aggregated over all matching anchors

link structure. To obtain vector object representations we turn towards the Linking Open Data cloud as it provides a richer source of structured data than Wikipedia. We use the 2009 version of the Billion Triple Challenge data set (BTC2009).

2.3 NILs

Handling entities which are absent from the knowledge base consists of two tasks: identifying them and subsequently clustering those that refer to the same (absent) entity.

2.3.1 NIL labeling

When the final ranking is produced, the last step is to map the Wikipedia entity ID to a TAC knowledge base entity ID. We do so in two steps: we first map the Wikipedia ID back to its title, and check for a literal match with any of the entities in the TAC knowledge base. If this does not return a match, we retrieve all redirect titles of the Wikipedia entity, as commonly titles that have changed over time are archived in this list. If both steps return no KB entity ID, we assume the entity to be a NIL.

2.3.2 NIL clustering

To cluster NIL entries, we convert the TAC KBP 2012 source documents to TF.IDF weighed vectors using the Gensim topic modeling framework for Python (Řehůřek and Sojka, 2010).

We then apply a hierarchical agglomerative clustering algorithm on the vectors of all documents that were labeled as NIL by our system. The clustering algorithm's cutoff is empirically determined with data from the TAC KBP 2011 track.

3 Experimental Setup

3.1 Machine learning

We use the random forest classifier of RT-Rank (Mohan et al., 2011) for our machine learning step. Random forest classifiers have proven to be best or near best in entity linking environments (Meij et al., 2012), and are robust to overfitting and to noise (Breiman and Schapire, 2001). The k parameter (the number of randomly selected features used for building each tree of a random forest) is set to 4 and

Table 3: Runs

Name	No. Features
Baseline	32
Context	40
Baseline + Context	72
Baseline + LOD	34
Baseline + Context + LOD	74

the number of trees is 1500.

3.2 Experiments

We measure the performance of using three approaches to this task: our baseline (BL) as described in 2.2.1, our context-aware extended feature (Context) as described in 2.2.2, and our LOD approach (LOD) as described in 2.2.3. To be able to determine how these different approaches interact, we extend the baseline with the context features, with the LOD features, with both the context and LOD feature sets, and finally we evaluate both the context features and baseline features in isolation. This results in the five runs described in table 3.

The B-cubed+ scoring metric is the official evaluation metric in the TAC KBP, since TAC KBP 2011 where the task of NIL clustering became a mandatory part of the task. This metric considers the system output as a collection of clusters, where queries linked to the same KB node are part of the same cluster. The B-cubed metric estimates the precision and recall for each item in a cluster, and uses this to calculate the average precision and recall for the complete set (Amigó et al., 2009).

4 Results and Analysis

At the time of submission we discovered a technical issue with the calculation of the features in our context featureset, which resulted in incoherent feature vectors. Furthermore, in the candidate generation phase, one of our steps produced duplicate candidates, introducing noise in our training data. In this section we present both the official results as submitted, and the results achieved after correcting for these errors, see table 4.

Our system does not perform well at NIL labeling. It predicted around 600-700 in each run, while there are in total 1049 queries referring to a NIL entity. This explains the large difference between the context featureset on the full query set and the in-KB subset. Furthermore, it explains why our erroneous submissions generally perform better at the full query set evaluation: because of noisy training data and degraded entity linking performance, more NILs were labelled. The NIL clustering improves overall performance when fewer entities are linked.

Table 4: B^3+ F1 Results

Full query set (2226)	official	corrected
Baseline (BL)	0.379	0.387
Context (Co)	0.428	0.427
BL+Co	**0.450**	0.434
BL+LOD	0.399	0.383
BL+Co+LOD	0.437	0.428
NIL subset (1049)	official	corrected
Baseline (BL)	0.388	0.398
Context (Co)	**0.648**	0.493
BL+Co	0.493	0.445
BL+LOD	0.446	0.399
BL+Co+LOD	0.469	0.434
In-KB subset (1177)	official	corrected
Baseline (BL)	0.364	0.370
Context (Co)	0.231	0.364
BL+Co	0.407	**0.418**
BL+LOD	0.351	0.361
BL+Co+LOD	0.402	0.415

As our NIL approach was identical across the five runs, the distinctions between runs in the in-KB subset provide more valuable insights. In this subset, combining the baseline with the context extension achieves highest performance. We believe this is caused by the datasets' ambiguity: single query mentions can refer to multiple entities, and multiple queries can refer to a single entity. Our baseline approach links identical mentions to the same entity, so it does not cope well with this ambiguity. This ambiguity is particularly high in the 2012 dataset: 41.35% of the in-KB queries are ambiguous, as opposed to 12.11% in the training data. The context feature in isolation does not involve the surface form of the query at all, which makes this approach miss useful local clues for disambiguation.

Compared to all other submissions (table 5), our best approaches perform under median. Only at the

Table 5: All submissions (24)
corrected result in brackets

Full queryset (2266)	B^3+ F1
Highest	0.730
Median	0.536
UvA BL+Co	0.450 (0.434)
NIL subset (1049)	B^3+ F1
Highest	0.847
Median	0.594
UvA Co	0.684 (0.493)
In-KB subset (1177)	B^3+ F1
Highest	0.687
Median	0.496
UvA BL+Co	0.407 (0.418)

official context-extended baseline submission, we achieve a NIL clustering F1 score of above median: 0.684.

5 Conclusions

We have shown that the performance of our baseline can be improved by incorporating features that analyze the context on the entity-level. It is likely that both the local and contextual approach to entity linking benefit from the different types of information that is considered. Furthermore we have seen that an erroneous system benefitted significantly from increased NIL labeling, this latter task deserves more attention in our future participation.

6 Acknowledgements

This research was partially supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 258191 (PROMISE Network of Excellence) and 288024 (LiMoSINe project), the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.061.814, 612.061.815, 640.004.802, 727.011.005, 612.001.116, HOR-11-10, the Center for Creation, Content and Technology (CCCT), the Hyperlocal Service Platform project funded by the Service Innovation & ICT program, the WAHSP and BILAND projects funded by the CLARIN-nl program, the Dutch national program COMMIT, by the ESF Research Network Program ELIAS, and Elite Network Shifts project funded by the Royal Dutch Academy of Sciences.

References

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.*, 12(4):461–486, August.

Leo Breiman and E. Schapire. 2001. Random forests. In *Machine Learning*, pages 5–32.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *EMNLP '07*, pages 708–716. ACL.

J. Heng, R. Grishman, and H. Dang. 2011. Overview of the tac2011 knowledge base population track. In *Proceedings of the Fourth Text Analysis Conference (TAC 2011)*.

E. Meij, W. Weerkamp, and M. de Rijke. 2012. Adding semantics to microblog posts. In *WSDM 2012*, pages 563–572, Seattle. ACM.

David Milne and Ian H. Witten. 2008. Learning to link with Wikipedia. In *CIKM '08*, pages 509–518, New York. ACM.

Ananth Mohan, Zheng Chen, and Kilian Q. Weinberger. 2011. Web-search ranking with initialized gradient boosted regression trees. *Journal of Machine Learning Research, Workshop and Conference Proceedings*, 14:77–89.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA. `http://is.muni.cz/publication/884893/en`.