UBC Entity Linking at TAC-KBP 2013: random forests for high accuracy

Ander Barrena, Eneko Agirre, Aitor Soroa
IXA NLP Group / University of the Basque Country, Donostia, Basque Country
abarrena014@ikasle.ehu.es, e.agirre@ehu.es, a.soroa@ehu.es

Abstract

This paper describe our systems and different runs submitted for the Entity Linking task at TAC-KBP 2013. We developed two systems, one is a generative entity linking model and the other is a supervised system reusing the scores of the previous model using random forests. Our main research interest is Named Entity Disambiguation task and we thus performed a very naive clustering of NIL instances. In fact, our best run scores at par to the best system on accuracy (ignoring NIL clustering), with another run we obtain top performance on KB mentions, both in accuracy and B-cubed F1.

1 Introduction

The Entity Linking task is the task of matching name mentions occurring in a document to a reference entity of a specific Knowledge Base (KB). For instance, in Figure 1 the mention "Lucy Walsh" is matched to the corresponding referent entity in Wikipedia. If there is no entity in the KB for a particular mention, it should be linked to the NIL entity. The Entity Linking task faces two main problems, *name variation* and *name ambiguity*. Name variation means that a single entity can be referred using different aliases, acronyms or even misspelled names. Name ambiguity problem arises from the fact that a single mention may refer to different entities in different contexts.

Our approach consist of three main steps. Given a query, we first search in the document for a name expansion of the given mention. Then comes the candidate generation step where we generate all possible candidates to the matched mentions. Finally our systems rank candidate entities for each mention. We did not perform any additional classification step for NIL entity, and we treated it as just another candidate to rank. We developed two different systems, which use the same name expansion and candidate generation algorithms, but with a different ranking step.

Figure 1: Entity Linking task example, the mention "Lucy Walsh" is linked to referent entity in Wikipedia.

The paper is structured as follows, first we are going to explain which are the resources we have used. Then we are going to analyze each main step mentioned before. Starting with name expansion, followed by candidate generation and ending with candidate ranking. Afterwards, results and conclusion will be presented.

2 Resources

We use a 2011 Wikipedia snapshot in our experiments. From the snapshot we extract two information resources: a dictionary and textual contexts for all candidate entities.

The dictionary is an association between strings and Wikipedia articles. We construct the dictionary using article titles, redirections, disambiguation pages, and anchor text. Mentions are lowercased and all text between parenthesis is removed. If the mention links to a disambiguation page, it is associated with all possible articles the disambiguation page points to. Each association between a string and article is scored with the prior probability, estimated as the number of times that the mention occurs in the anchor text of an article divided by the total number of occurrences of the mention. Note that our dictionary can disambiguate any mention, just returning the article with highest score.

We also extract textual contexts for all the possible candidate entities (see below). Given an entity, we collect all the mentions to this entity within Wikipedia, and extract a context of 50 words around the anchor link. Contexts are lemmatized and POS tagged using the Stanford CoreNLP suite[1].

3 Name expansion

The first step matches the query mention in the document. One of the problems we have seen during development was that many times the query mention is too short. For example, Figure 2 shows one query from the 2012 TAC dataset, whose query mention is "Lucy". According to our dictionary, Wikipedia has more than 190 entities linked to the name "Lucy", so the disambiguation step will be really hard. Following usual practice, we search in the given document for larger string names if available. In the example "Lucy Walsh" is a proper entry in the dictionary and it is linked to a single entity[2] which happens to be the correct one. We think that this is an important step which significantly improves the system results.

4 Candidate Generation

The second step generates candidate entities for the matched mentions by just assigning all entities linked to the mention in the dictionary. If the matched mention does not exist in the dictionary, we apply a *Did you mean* (DYM) algorithm[3] to correct possible misspellings. This way, misspelled strings like "Lucy Walhs" are correctly handle, improving our candidate generation capacity. Note that we need to access the web in order to perform the DYM algorithm. Finally, we also add the NIL entity as a possible candidate for each mention.

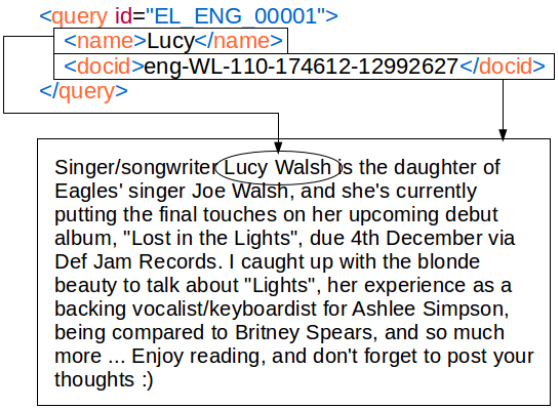

Figure 2: Example of name expansion in TAC-KBP 2012 dataset sample.

5 Candidate Ranking

We developed two systems to rank candidate entities. Our systems ranks the NIL entity as any other entity, so we did not use an external classifier to do that work.

5.1 iXa

iXa is a generative entity linking model based on (Han and Sun, 2011), where candidate entities are ranked combining evidences from 3 different probability distributions, which we call entity knowledge, name knowledge and context knowledge, respectively.

Entity knowledge $P(e)$ represents the probability of generating entity e, and is estimated as follows:

$$P(e) = \frac{Count(e) + 1}{|M| + N}$$

where $Count(e)$ describes the entity popularity, e.g., the number of times the entity e is referenced

[1] http://nlp.stanford.edu/downloads/corenlp.shtml

[2] http://en.wikipedia.org/wiki/Lucy_Walsh

[3] http://en.wikipedia.org/w/api.php

within Wikipedia, $|M|$ is the mention size and N is the total number of entities in Wikipedia. As can be seen, the estimation is smoothed using the *add-one* method.

Name knowledge $P(s|e)$ represents the probability of generating a particular string s given the entity e, and is estimated as follows:

$$P(s|e) = \frac{Count(e,s)+1}{Count(e)+S}$$

where $Count(e,s)$ is the number of times mention s is used to refer entity e and S is the number of different possible names used to refer to e.

Finally the context knowledge $P(c|e)$ represents the probability of generating context $c = \{t_1, t_2, \ldots, t_n\}$ given the entity e, and is estimated as follows:

$$P(c|e) = P_e(t_1)P_e(t_2)...P_e(t_n)$$

where $P_e(t)$ is estimated as:

$$P_e(t) = \lambda P'_e(t) + (1-\lambda)P_g(t)$$

$P'_e(t)$ is the maximum likelihood estimation of each term t in the context of e entity. Context words are smoothed by n-gram frequency (Jelinek and Mercer, 1980)[4]. λ parameter is set to 0.2 according to (Han and Sun, 2011).

This framework integrates the NIL detection in an uniform way. We consider NIL as a extra entity, which has the following distributions based on the same n-gram counts as used for smoothing:

$$P(\text{NIL}) = \frac{1}{|M|+N}$$
$$P(s|\text{NIL}) = \prod_{t \in s}^{n} P_g(t)$$
$$P(c|\text{NIL}) = \prod_{t \in c}^{n} P_g(t)$$

Finally, we combine all evidence to find the entity that maximizes the following formula:

$$e = \arg\max_e P(s,c,e) =$$
$$\arg\max_e P(e)P(s|e)P(c|e)$$

In development experiments, we did not manage to replicate the performance reported by the authors (Han and Sun, 2011), which motivated us to explore supervised methods to combine the models mentioned above.

5.2 iXa-RF

iXa-RF is our second system developed for entity linking task and is based on previous model. The main idea of iXa-RF is to use the probabilities given by the previous model as features for a supervised classifier. Specifically, we create an instance for each query mention and candidate pair, and attach the following features to it:

- $P(e)$,$P(s|e)$ and $P(c|e)$.
- $P(e)P(s|e)$.
- $P(e)P(s|e)P(c|e)$.

Furthermore we use the difference and percentage scores between candidate probabilities for the same query mention as additional features. We also include features mentioned in (Paul McNamee, 2010) based in name similarity and document analysis.

In addition, we also consider an extra feature based on random walks over the Wikipedia link structure. This weight represents, loosely speaking, the relative importance of the entity given the mention and its surrounding context.

The task of the classifier is then to choose the best candidate for each query mention, i.e., the instance with lower classification error among all instances of a query mention. We used a random-forest classifier comprising 101 decision trees.

6 Knowledge base Mapping

Our systems return entities from the 2011 Wikipedia snapshot, and we need to link them to the TAC-KBP knowledge base[5]. If there is no direct match, we test whether there is any reference KB entity which redirects to the entity returned by the system, according to the 2011 version of Wikipedia. If so, we return the KB entity and if not we return NIL.

In addition to the strategy above, we also set some runs to only rank KB entities. This way, we reduce the number of candidates.

[4]We used Google Web 1T corpus (T. Brants and A. Franz, 2006) for frequency counts.

[5]The reference KB for TAC is a subset of a 2008 dump of Wikipedia.

7 NIL clustering

The systems return NIL in three cases:

- There is not candidate for the mention according to our dictionary.
- NIL is the highest scoring entity.
- The entity returned by the system does not map to the reference KB.

We perform a very basic clustering for NIL entities. All mentions linked to the NIL entity having the same query string are clustered in the same group.

8 Experimental results

As said before (Section 2), we used a 2011 Wikipedia dump to build the dictionary and entity contexts. The supervised algorithm is trained using all previous TAC-KBP datasets available[6].

We submitted a total of 5 runs, all of them accessing to Internet at some point (needed by the DYM algorithm described in Section 3). None of our runs uses wiki text, so we submitted the same five runs to the official track and to the *without wiki text* track.

Here comes the explanation of each run:

- **dict**: disambiguates each query mention returning the highest scoring entity according to dictionary (see Section 2). We use this system as a baseline. Run number 3.
- **iXa**: generative entity linking model (see Section 5.1). Run number 4.
- **iXa-KB**: same as *iXa*, but only considering the entities which can be mapped to the reference KB and ignoring the rest (see Section 6). Run number 5.
- **iXa-RF**: supervised machine learning without using the random walk feature (see Section 5.2). We only consider entities which can be mapped to the reference KB. Run number 1.
- **iXa-RF-RW**: same as *iXa-RF*, but including the whole feature set. Run number 2.

[6] 2009 test, 2010 train and test, 2011 test and 2012 test.

Table 1 shows the F1 measure for Bcubed+ score that our runs get in TAC-KBP entity linking 2013. Our best run, **iXa-RF-RW**, gets 0.642 F1 score for all queries. According to summary statistics released by NIST, this value is 8 points over the median but 10 points below from best score. We did not done any especial effort in NIL clustering and we did not obtain very good results in NIL F1 measure. But taking into account that our research is focused on named-entity disambiguation, KB values show that our systems have done a good work. All runs score close to the best results (in KB column) according the summary statistics. Ranking only KB entities with a generative entity linking model, **iXa-KB**, we reach our best value in KB.

Run	All	in KB	NIL
dict	0.611	0.672	0.518
iXa	0.619	0.676	0.531
iXa-KB	0.565	**0.714**	0.369
iXa-RF	0.631	0.671	**0.566**
iXa-RF-RW	**0.642**	0.689	**0.566**
Best	0.746	0.722	0.777
Median	0.560	0.537	0.575

Table 1: Bcubed+ F1 measure for our submitted runs compared to best and median performance.

The accuracy-based evaluation does not take into account the NIL clustering. Table 2 shows the accuracy score that our runs get in TAC-KBP 2013. Our supervised machine learning algorithm including the whole feature set, **iXa-RF-RW**, gets a 0.826 score for all queries, and this is very close to the value of the best system this year. Our dictionary is able to disambiguate correctly 74% of queries without context information, as shown in run **dict**. Finally **iXa-KB** run, gets a 0.783 score in KB queries, very close to the best in KB result.

9 Conclusions and future work

Our approach focuses in KB queries and we obtain very good results in this field. We have seen that using the distribution probabilities as features in a supervised machine learning algorithm, the results improve considerably.

In the future we plan to build a clustering system to improve our NIL results.

Run	All	in KB	NIL
dict	0.775	0.740	0.816
iXa	0.789	0.746	0.840
iXa-KB	0.673	**0.783**	0.543
iXa-RF	0.817	0.737	**0.912**
iXa-RF-RW	**0.826**	0.752	**0.913**
Best	0.833	0.788	—
Median	0.720	0.614	—

Table 2: Accuracy measure for our submitted runs compared to best and median performance.

Acknowledgements

The research leading to these results was carried out as part of the READERS project (`http://nlp.uned.es/readers-project/`) funded by European Communitys in the framework of ERA-NET CHIST-ERA. The work has been also funded by the Basque Government (project IBILBIDE, SAIOTEK S-PE12UN089).

References

Xianpei Han and Lee Sun. 2011. *A generative entity-mention model for linking entities with knowledge base*. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1. Stroudsburg, PA, USA.

Chang, Angel X. and Spitkovsky, Valentin I. and Yeh, Eric and Agirre, Eneko and Manning, Christopher D. 2010. *Stanford-UBC Entity Linking at TAC-KBP*. Proceedings of the Third Text Analysis Conference (TAC 2010). Gaithersburg, Maryland, USA.

Frederick Jelinek and Robert L. Mercer 1980. *Interpolated estimation of Markov source parameters from sparse data.*. Proceedings of the Workshop Patter recognition in practice, pages 381-397.

T. Brants and A. Franz. 2006. *Web 1T 5-gram corpus version 1.1*. Technical report, Google Research.

Paul McNamee. 2010. *HLTCOE Efforts in Entity Linking at TAC KBP 2010*. Proceedings of the Third Text Analysis Conference (TAC 2010).