Supervised Learning of Entity Disambiguation Models by Negative Sample Selection

Hani Daher, Romaric Besançon, Olivier Ferret, Hervé Le Borgne, Anne-Laure Daquo, and Youssef Tamaazousti

CEA, LIST, Vision and Content Engineering Laboratory, Gif-sur-Yvette, France.
{hani.daher,romaric.besancon,olivier.ferret,herve.le-borgne, anne-laure.daquo,youssef.tamaazousti}@cea.fr

Abstract. The objective of Entity Linking is to connect an entity mention in a text to a known entity in a knowledge base. The general approach for this task is to generate, for a given mention, a set of candidate entities from the base and determine, in a second step, the best one. This paper focuses on this last step and proposes a method based on learning a function that discriminates an entity from its most ambiguous ones. Our contribution lies in the strategy to learn efficiently such a model while keeping it compatible with large knowledge bases. We propose three strategies with different efficiency/performance trade-off, that are experimentally validated on six datasets of the TAC evaluation campaigns by using Freebase and DBpedia as reference knowledge bases.

1 Introduction

In the domain of Information Extraction, the Entity Disambiguation task (or Entity Linking) consists in connecting an entity extracted from a text to known entities in a knowledge base [11, 17], which is useful for further extraction tasks (relation extraction or event detection, for instance) or to provide a unique normalization of the entities in an Information Retrieval context. This task is sometimes part of a more general framework that globally disambiguates all the concepts in a document with respect to a knowledge base, whether they are named entities or nominal expressions (e.g. Wikify [12] or Babelfy [13]).

An Entity Disambiguation system is usually based on three main steps [8]. First, it analyzes an input (query) to identify an "entity mention" that needs to be linked to the knowledge base. Second, for each mention, the system generates several candidate entities from the knowledge base and finally, it selects the best entity among the candidates. For such systems, one of the main challenge is to deal with the very large number of entities present in the knowledge base.

The main contribution of this paper focuses on the last step of this process: given a set of candidates for an entity mention, we propose a new method to select the best one. The main idea is to compute a model that allows to discriminate a given candidate from its most ambiguous entities: this model will then give a score measuring the association between the entity mention and each candidate entity, that we call the *Discriminative Disambiguation Score*. Since our disambiguation model must be learned for each entity of the knowledge base, and due to the large number of these entities (several millions), we adopted a linear model, which complexity can be supported on a large scale, both for training and during the (possibly online) testing phase. Hence, the core of our contribution lies in the strategy for building the training datasets: in particular, choosing the right set of negative examples allows us to manage several millions of models in a tractable time for training. We evaluate this model using Freebase and DBpedia as reference knowledge bases and six test datasets from the TAC evaluation campaigns.

2 Related Work

One way of classifying the different approaches for Entity linking is the degree of supervision they required. Unsupervised methods generally rely on the definition of a similarity score between the entity mention and the entity in the knowledge base: selecting the correct entity simply corresponds to maximizing this score. Such similarity scores are usually based on the overlap of contexts [4] and can combine several measures: for instance, Han & Zhao [7] combine similarities between words and Wikipedia concepts. These methods are usually simple and easy to implement but they have a lower performance, when compared to the supervised methods as it was shown in past evaluation campaigns [3].

Supervised methods are generally based on binary classifiers [10, 18] or ranking models [16, 2], specifically dedicated to entity disambiguation. In both cases, the difficulty lies in building labeled data, which is time consuming, especially for large knowledge bases like Freebase or DBpedia. Among these approaches, some studies [20, 6] use ambiguous entities to learn the models that allow to select the correct entity but, as far as we know, none of them proposes to build a discriminative model for each entity. Zhang *et al.* [20] focuses on the unambiguous mentions of entities in DBpedia to automatically create training examples for their disambiguation model. idea is to generate disambiguation examples by replacing in documents the mentions of an entity that are not ambiguous by alternate names of this entity that are ambiguous. The positive examples are built by associating the modified documents with the entity while the negative examples are produced by associating these documents with the entities referred by the alternate names. Zhang *et al.* [19] uses an iterative learning algorithm to select the most informative entities that are close to the separating hyperplane. Fan *et al.* [6] uses a one-vs-all strategy to disambiguate entities. Instead of training one classifier per entity in Freebase, they propose a strategy to merge all of them into one generic classifier, which consists in using, for an entity, its unique Freebase identifier as an extra descriptor. All the positive examples of a given entity have thus the same Freebase identifier. The negative examples are randomly chosen from entities with a similar name.

Finally, some studies use a semi-supervised approach, such as [21], that address the problem of data acquisition and labeling, using a two sets of labeled and unlabeled data. In an iterative approach, a model is learned using positive examples extracted from Wikipedia pages that contain an unambiguous entity (thus providing reference data) and negative examples taken randomly from other entities. The learned model is used to annotate the unlabeled documents, which will then be used in the learning phase of the next iteration.

3 Discriminative Disambiguation Score

In the context of supervised systems, we present in this section the new score we propose for Entity Linking, named Discriminative Disambiguation Score (DDS). The idea of DDS is to reflect the likelihood of an entity mention to be disambiguated by a given candidate entity. It represents the posterior probability $P(candidate_i|mention)$ that a candidate is appropriate to disambiguate a given entity mention, computed from a classifier score [15].

The novelty of our proposal is to learn a classifier for each candidate entity (as long as the required data are available). Such an approach has sometimes been dismissed and considered impossible because of intractable computational issues [6]. Indeed, one difficulty of our approach lies in the capacity of learning such classifiers for several millions of entities while still keeping a relevant discriminative power for each of them. First, for the sake of computational tractability, both at learning and testing time, we restrain our approach to linear classifiers (in practice, we used logistic regression models, but we obtained comparable results with SVMs, that are not reported in this paper).

For each candidate, we must select both positive and negative examples, extract features then learn the classifier. In all cases, the vector representation of examples is based on a *tf-idf* model relying on the same vector space, built from the complete collection of the Wikipedia pages associated with the entities in the knowledge base. Regarding the positive examples, the textual context of each entity in the knowledge base is considered by using the following information:

– Abstract of the Wikipedia page associated with the entity;
– Paragraphs explicitly containing the entity in the Wikipedia page of the entity;
– Paragraphs, from other Wikipedia pages, that contain a wikilink pointing to the entity.

As suggested by [6] a direct one-versus-all strategy would be computationally intractable. In order to solve this problem we propose three approaches to select a subset of negative examples that contains representative *tf-idf* vectors.

– **DDS-Rand**: In the Random approach, the negative examples are randomly selected from the positive examples of all the other entities in the knowledge base. There is no constraint on whether the negative examples should only be selected from ambiguous entities.
– **DDS-Ambig**: In the Ambiguous approach, the negative examples are randomly selected from the positive examples of the ambiguous entities. The ambiguous entities are generated by using the same approach as for the candidate entity generation (see Section 4.1): for each known form of the entity

in the knowledge base (normalized form or variation), ambiguous entities are the entities that share a common form or have a close form (inclusion or string edit distance ≤ 2). Since the set of ambiguous entities can be large, the negative examples are actually selected from a random subset.
- **DDS-Ambig-NN**: In the Ambiguous Nearest Neighbor approach, the entity for which we want to compute a discriminative model is represented by the centroid of the *tf-idf* vectors that constitute its positive examples. We then select as negative examples the *tf-idf* vectors that have the closest Cosine similarities to this representation among all the examples from the ambiguous entities. These negative examples are considered as the most informative data instances and the most relevant for discrimination because they are the most ambiguous with the entity.

4 Overall Entity Linking System

In order to test the Discriminative Disambiguation Score (DDS), we integrate it in a global Entity Linking system that relies on a supervised learning framework: the disambiguation of an entity mention is performed by a trained classifier based on a set of features. The DDS is added as a particular feature in the system. The Entity Linking system uses a standard architecture [8] composed of two main steps: for a given entity mention and its textual context, a first module generates possible candidate entities for the linking and second one takes as input the different candidate entities and select the best one.

4.1 Generation of Candidate Entities

The generation of the candidate entities relies on both the analysis of the entity mention and its textual context. In this study, we focus mainly on the disambiguation of entities, not their recognition. Therefore, the entity mentions to disambiguate are given as input to the system. A complementary analysis of these entity mentions in the text is carried out, in order to associate a type (Person, Location, Organization) with the entity mentions[1] and define their context in terms of surrounding entities (we consider only the explicit named entity mentions and we ignore the nominal and pronominal mentions). Two forms of entity mention expansion are performed, which can be considered as simple coreference approaches: (i) if an entity mention is an acronym, we search the text for entity mentions of the same type whose initials correspond to the acronym (ii) we search the text for other entity mentions whose expression includes the target entity mention as a substring. These other forms are added as variations for the entity mention.

After the intrinsic analysis of the entity mention, candidate entities are generated by comparing the entity mention with the entities of the knowledge base, using the following four strategies [5]:

[1] We used the tool MITIE for this step (`https://github.com/mit-nlp/MITIE`).

- Equality between the forms of the entity mention and an entity in the knowledge base;
- Equality between the form of the entity mention and a variation (alias or translation) of an entity in the knowledge base;
- Inclusion of the form of the entity mention in one of the forms of the variations of an entity in the knowledge base;
- Similarity between the form of the entity mention and a variation of an entity in the knowledge base. We use the Levenshtein distance, which is well suited to overcome the spelling errors and name variations. In the experiments, we considered an entity in the knowledge base as a candidate entity if its form or any of its variations have a distance with the form of the entity mention ≤ 2. For better efficiency, we exploited a BK-tree structure [1] for this selection.

The candidate entities are also filtered in order to keep only entities that have one of the expected named entity types (e.g. Person, Location, Organization).

4.2 Selection of the Best Candidate Entity

The goal of this step is to find the correct candidate entity in the set of generated candidate entities. To this purpose, a classifier is trained to recognize the best entity among the entity candidates, using training data composed of disambiguated entity mentions. More precisely, each candidate entity is associated with a set of features:

- Four binary features indicating which strategy was used for the generation of this candidate entity;
- Two general scores comparing the context of the entity mention with the context of the candidate entities in the knowledge base. The first score focuses on their lexical context. It compares, with the Cosine similarity, a vector representation of the textual context of the entity mention (we considered the whole document as the context of the entity mention in these experiment) and the vector representation of the Wikipedia page of the candidate entity. The second score focuses on a context based on surrounding entities and compares, once again with the Cosine similarity, the vector representation of the textual context of the entity mention[2] and a vector representation of the entities in relation with the candidate entity in the knowledge base. The vector space supporting these representations is built from all the Wikipedia pages of the entities in the knowledge base;
- One of the DDS scores computed as presented in Section 3.

A binary classifier is then trained to associate such a set of features with a decision whether the candidate entity is the correct one for the entity mention. Using the training data, we generate the candidate entities from the entity mentions: the positive examples for the training are then formed by the (entity

[2] For the entity mention, we took the whole lexical context as we did not have an entity recognizer for all the entity types of the knowledge base.

mention, candidate) pairs that correspond to the expected link in the reference. The negative examples are pairs with wrong candidates generated for the entity mentions. Since the classes are imbalanced (the number of candidates generated for each query may be high, between 1 and 460 in our experiments), we use undersampling by limiting the number of negative examples to 10 times the number of positive examples. Each decision of the classifier is weighted by the probability estimate of the classifier and the candidate entity with the highest probability is selected as the final disambiguated entity. In the standard entity disambiguation task, the system must also be able of determining when an entity mention does not link to any entity in the knowledge base (these are referred as NIL entities). In our approach, this occurs if no candidate is generated or if all candidates are rejected by the classifier.

5 Experiments and Analysis

5.1 Datasets

To validate our approach, we use the 2009-2013 and 2015 datasets of the TAC-KBP evaluation campaign. For TAC 2015, we consider the monolingual English Diagnostic Task (where the entity mentions in the query texts are already given as input), in order to use the same evaluation framework as for the other datasets. We report in Table 1 the main features of these datasets. For the 2009-2013 campaigns, the reference knowledge base is extracted from Wikipedia infoboxes (similarly to DBpedia) [14]. It contains 818,741 entities, which are all associated with a Wikipedia page. In the 2015 campaign, the knowledge base was built from Freebase [9]. The whole Freebase snapshot contains $43M$ entities but a filter is applied to remove some entity types that are not relevant to the campaign (such as music, book, medicine and film), which reduces it to $8M$ entities. Among them, only 3,712,852 (46%) have an associated content in Wikipedia and are thus subject to provide positive examples to learn the DDS. In the 2015 campaign, the purpose was to link all the entities of a restricted set of documents. On the contrary, the former campaigns aimed at linking a restricted number of entity per document; hence, the number of entities and documents is approximately the same for the 2009-2013 campaigns.

Table 1. Description of the datasets used in the evaluation process

	Dbpedia			**Freebase**	
	Nb. docs.	**Nb. entities**		**Nb. docs.**	**Nb. entities**
TAC 2009	3,688	3,904	**TAC 2015 train**	168	12,175
TAC 2010	2,231	2,250	**TAC 2015 test**	167	13,587
TAC 2011	2,231	2,250			
TAC 2012	2,016	2,226			
TAC 2013	1,820	2,190			

Table 2. Candidate statistics for the DBpedia and Freebase datasets (TAC 2009-2013 and 2015)

Dataset	Nb queries	NIL queries	Nb cand.	NIL cand.	Avg. cand.	cand. recall
Dbpedia						
2009	3,904	2,229	208,060	949	70.41	84.0%
2010	2,250	1,230	232,672	601	141.10	89.4%
2011	2,250	1,126	329,508	388	176.96	87.9%
2012	2,226	1,049	420,179	117	199.23	92.4%
2013	2,190	1,007	394,217	395	219.62	83.5%
Freebase						
2015 train	12,175	3,215	5,844,592	1,282	458.08	76.0%
2015 test	13,587	3,379	6,141,369	1,255	480.32	77.6%

5.2 Results on Candidate Generation and DDS

Generated Candidate Entities. We present in Table 2 some statistics on the queries and the generated candidate entities. In particular, the candidates recall, defined by the percentage of non-NIL queries for which the expected candidate is in the candidate list, is quite good for the 2009-2013 datasets, using simple candidate generation strategies that generate a reasonable number of candidates per query (150 in average). For the 2015 dataset, the KB contains 10 times more entities and the number of generated candidates is much larger. In addition, the candidate recall is lower (77%): an analysis of the missing entities showed that the variations contained in the KB should be enriched for a better coverage (e.g., links between nationalities and countries are missing, such as French $\rightarrow$ France).

Discriminative Disambiguation Score. The extraction of the textual context (Section 3) from the Wikipedia pages is performed for the 818,741 entities in DBpedia and the 3,712,852 entities in Freebase. A vector space model of 169,647 dimensions, built from the whole Wikipedia dump, is used to convert these set of paragraphs into *tf-idf* vectors. When applied on DBpedia, a total number of 32,939,218 examples are generated. On average, an entity has around 40.18 examples and between 1 and 119,178 examples per entity. When applied on Freebase, a total number of 97,157,120 examples are generated. On average, an entity has around 26.16 examples and the number of examples per entity is between 1 and 119,197. The candidate entities for TAC 2009-2013 (Table 1, Nb. Candidates) represent 41,313 unique entities in DBpedia. For Freebase, the candidate entities that are associated with a Wikipedia page represent 124,456 unique entities, cumulated on training and test datasets. For each candidate entity in DBpedia and Freebase, we train a classifier based on the approaches described in Section 3: in our experiments, we used a L2-regularized logistic regression classifier, from the LIBLINEAR library[3], whose complexity is $O(n)$, where n is the number of features.

[3] `https://www.csie.ntu.edu.tw/~cjlin/liblinear/`

Table 3. Minimum, Maximum and Average time (in seconds) needed for each approach to select the negative examples and train the classifier on DBpedia and Freebase

	Dbpedia			**Freebase**		
	Min.	**Max.**	**Avg.**	**Min.**	**Max.**	**Avg.**
DDS-Rand	0.003	109.59	1.55	0.002	49.29	1.13
DDS-Ambig	0.01	398.47	11.65	0.006	270.45	6.49
DDS-Ambig-NN	0.027	2551.62	146.88	0.014	2102.31	85.49

Table 4. Cross-validation results of the classifiers trained on 26,819 datasets having each at least 100 positive examples

	Precision	**Recall**	**F-score**
DDS-Rand	0.987 ± 0.015	$0,969 \pm 0.042$	0.977 ± 0.028
DDS-Ambig	0.963 ± 0.050	$0,919 \pm 0.112$	0.937 ± 0.086
DDS-Ambig-NN	0.954 ± 0.058	$0,798 \pm 0.188$	0.857 ± 0.151

We report in Table 3 the minimum, maximum and average time in seconds needed to train a classifier for the different approaches. *DDS-Rand* is the simplest approach in complexity. This is why it needs less time to select the negative examples and train the classifier. *DDS-ambig-NN* takes more time because of the negative examples selection module where we have to compute the distance between the centroid of the entity and each *tf-idf* vector of its ambiguous entities.

In order to test the relevance of the DDS scores, we first selected a subset of entities from the DBPedia KB that have at least 100 positive examples and evaluated the performance of the trained classifiers for these entities using a 5-fold cross validation on this subset. The accuracy results of the classifiers trained on these 26,819 entities are reported in Table 4 and show that these classifiers achieve a good performance in differentiating a particular entity from the others. The results are a bit lower for *DDS-ambig-NN* because, in this case, we specifically selected the closest negative examples, which makes the disambiguation task harder.

5.3 Entity Linking Results

In this section, we present the results of the DDS scores in a full entity linking system. We compare the results obtained by the Baseline system (as presented in Section 4.2, without the DDS) with the results obtained when adding the DDS feature (this DDS feature is the score given by the classifier of the considered candidate entity for the classification of the entity mention). In order to verify the interest of the discriminative models compared to the simple addition of more textual contexts for the entity, we also considered a score computed as the Cosine similarity between the centroid of the positive examples of a candidate entity and the textual context of the entity mention. We name this score DDS-baseline.

Training and Testing datasets. For the TAC 2009-2013 datasets, no training data was provided. Therefore, we used for each year the data from the other years as training data.

Evaluation Measures. We use standard precision/recall/F-score measures on three criteria: the correct recognition of the reference entity when it exists (*link*), the correct recognition of an entity without a reference entity (*nil*) and the combined results (*all*). These measures correspond to the measures named *strong_link_match, strong_nil_match and the strong_all_match* in the TAC 2015 evaluation campaign [9]. We did not take into account the type of the named entities in this evaluation.

Results and Analysis. Table 5 reports for each approach the F-score for respectively the *strong_nil_match, strong_link_match and the strong_all_match* evaluation measures. These results show that including the DDS score to the set of features used for the entity disambiguation clearly improves results. The best results are obtained with the *DDS-Ambig-NN* for the TAC 2009-13 datasets, whereas *DDS-Ambig* gives the best results for TAC 2015. Interestingly, we note that even if *DDS-Ambig-NN* does not perform as well as the other approaches in the pure classification task, its usage in full system is beneficial. It tends to show that this model learns more discriminant information that better complements the information given by the other features.

Table 5. F-score results obtained with the addition of DDS scores. We report the strong_nil_match (top, best in italic), strong_link_match (middle, best in italic), strong_all_match evaluation (bottom, best is bold) criteria on TAC datasets

		2009	**2010**	**2011**	**2012**	**2013**	**2015**
	nil	0.851	0.863	0.808	0.649	0.801	0.668
Baseline	*link*	0.707	0.743	0.645	0.441	0.717	0.588
	all	0.795	0.813	0.735	0.533	0.761	0.601
	nil	0.851	0.859	0.807	0.649	0.8	0.667
DDS-Baseline	*link*	0.709	0.736	0.639	0.446	0.705	0.603
	all	0.796	0.808	0.734	0.535	0.754	0.611
	nil	0.856	0.858	0.817	*0.651*	0.801	0.679
DDS-Rand	*link*	0.72	0.751	0.646	0.436	0.704	*0.659*
	all	0.803	0.813	0.741	0.531	0.753	0.654
	nil	0.858	0.867	0.812	0.643	0.799	*0.694*
DDS-Ambig	*link*	0.73	0.762	0.647	0.454	0.722	0.654
	all	0.808	0.824	0.741	0.537	0.763	**0.656**
	nil	*0.874*	*0.884*	*0.821*	0.649	*0.82*	0.687
DDS-Ambig-NN	*link*	*0.754*	*0.796*	*0.663*	*0.468*	*0.756*	0.644
	all	**0.828**	**0.848**	**0.752**	**0.547**	**0.789**	0.646

Table 6. F-score results obtained using *only* the DDS as a disambiguation feature. We report the strong_nil_match (top, best in italic), strong_link_match (middle, best in italic), strong_all_match evaluation (bottom, best is bold) criteria on TAC datasets

		2009	**2010**	**2011**	**2012**	**2013**	**2015**
DDS-Baseline	*nil*	0.749	0.718	0.655	0.639	0.634	0,405
	link	0.289	0.222	0.278	0.22	0.182	0,002
	all	0.609	0.56	0.493	0.489	0.47	0,245
DDS-Rand	*nil*	0.828	0.838	*0.771*	0.818	0.757	0.611
	link	0.622	0.687	*0.546*	0.156	0.585	*0.508*
	all	0.749	0.772	**0.67**	0.537	0.672	**0.541**
DDS-Ambig	*nil*	0.815	0.833	0.662	0.565	0.748	*0.665*
	link	0.568	0.666	0.243	0.221	0.56	0.443
	all	0.73	0.768	0.481	0.395	0.667	0.517
DDS-Ambig-NN	*nil*	*0.84*	*0.849*	0.756	*0.754*	*0.772*	0.633
	link	*0.625*	*0.703*	0.429	*0.446*	*0.614*	0.433
	all	**0.771**	**0.797**	0.645	**0.638**	**0.708**	0.503

In order to further assess the influence of the DDS, we tested a disambiguation approach using only this score: we trained a classifier for the final disambiguation using the DDS as single feature (which allows learning automatically a threshold on the DDS). Table 6 reports the results obtained with this method, for each variant of DDS. With the exception of the 2011 and 2015 evaluation campaigns, the *DDS-Ambig-NN* score used as single feature produces the best results and is often not very far from the results obtained by the Baseline approach. For the 2012 dataset, the DDS alone produces a *strong_all_match* F-score of 63.8%, which is even better than the one obtained when combined with the other features.

Table 7. Ranking of our approaches compared to the official results of the campaigns, based on the strong_all_match evaluation criterion.

	2009	**2010**	**2015**
Nb. of teams	18	21	10
Median	0.67	0.683	0.634
Min.	0.0085	0.345	0
Max.	0.822	0.864	0.875
Baseline	0.783 (8)	0.798 (9)	0.601 (5)
DDS-Baseline	0.791 (6)	0.81 (8)	0.611 (5)
DDS-Rand	**0.803 (2)**	**0.821 (3)**	0.654 (4)
DDS-Ambig	0.794 (5)	0.815 (6)	**0.656 (4)**
DDS-Ambig-NN	0.795 (3)	0.82 (4)	0.641 (4)

Finally, we present a comparison of the results we obtained with the other teams that participated in the TAC evaluation campaigns in Table 7. We only show the results for 2009, 2010 and 2015 because the official measure for these years – the micro-average KB accuracy – corresponds to the measure we use in this paper. For 2011-2013, the official measure is the $B^3 + F1$ score, which requires a clustering of the NIL entities that we do not perform here. Note as well that the official scores in 2015 takes into account the type of the entity while we only consider the accuracy of the disambiguation. For the 2009 and 2010 datasets, the *DDS-Rand* approach achieves results that are not far from the best participant: it would be ranked respectively 2^{nd} out of 18 participants and 3^{rd} out of 21 participants[4]. For the 2015 dataset, our system still needs some tuning on the candidate generation to obtain better candidate recall but the general trend of the results provides solid ground to indicate that the DDS is a good feature for entity disambiguation.

6 Conclusion

We proposed a new feature for the entity disambiguation task based on a supervised approach to learn discriminative models for each entity in a knowledge base. By combining this feature to a set of features commonly used in the literature, the scores, expressed as percentages, increased by more than 4 points. Our entity disambiguation method showed its stability when it was tested on several Entity Linking evaluation campaigns, using two different knowledge bases. We also addressed the problem of the selection of negative examples by proposing three approaches. The *DDS-Rand* and *DDS-Ambig* approaches provide improvements over our baseline system with a very low computational time and a linear complexity. *DDS-Ambig-NN* gives better results but with a higher computation time. Most importantly, we showed that individual binary classifiers can be trained for each entity of a large knowledge base for a disambiguation task. We plan to improve the performance of the DDS by using dense vector representations (word embeddings) to represent the positive and negative examples, this type of vector representations having proven their efficiency for various classification tasks.

References

1. Burkhard, W.A., Keller, R.M.: Some Approaches to Best-match File Searching. Communications of the ACM 16(4), 230–236 (Apr 1973)
2. Cao, Z., Tao, Q., Tie-Yan, L., Ming-Feng, T., Hang, L.: Learning to Rank: From Pairwise Approach to Listwise Approach. In: 24th International Conference on Machine Learning (ICML 2007). pp. 129–136. Corvalis, Oregon, USA (2007)
3. Cassidy, T., Chen, Z., Artiles, J., Ji, H., Deng, H., Ratinov, L.A., Zheng, J., Han, J., Roth, D.: CUNY-UIUC-SRI TAC-KBP2011 Entity Linking System Description. In: Text Analysis Conference (TAC 2011) (2011)
4. Cucerzan, S.: Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In: 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007). pp. 708–716. Prague, Czech Republic (2007)

[4] The comparison is not absolutely fair since we used the data from other years for training, which were not available to the participants.

5. Dredze, M., McNamee, P., Rao, D., Gerber, A., Finin, T.: Entity Disambiguation for Knowledge Base Population. In: 23rd International Conference on Computational Linguistics (COLING'10). pp. 277–285. Beijing, China (2010)
6. Fan, M., Zhou, Q., Zheng, T.F.: Distant Supervision for Entity Linking. In: 29th Pacific Asia Conference on Language, Information and Computation (PACLIC 29). pp. 79–86. Shanghai, China (2015)
7. Han, X., Zhao, J.: NLPR_KBP in TAC 2009 KBP Track: A Two-Stage Method to Entity Linking. In: Text Analysis Conference (TAC 2009) (2009)
8. Ji, H., Nothman, J., Hachey, B.: Overview of TAC-KBP2014 Entity Discovery and Linking Tasks. In: Text Analysis Conference (TAC 2014) (2014)
9. Ji, H., Nothman, J., Hachey, B., Florian, R.: Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking. In: Text Analysis Conference (TAC 2015) (2015)
10. Lehmann, J., Monahan, S., Nezda, L., Jung, A., Shi, Y.: LCC Approaches to Knowledge Base Population at TAC 2010. In: Text Analysis Conference (2010)
11. Ling, X., Singh, S., Weld, D.: Design Challenges for Entity Linking. Transactions of the Association for Computational Linguistics (TACL) 3, 315–328 (2015)
12. Mihalcea, R., Csomai, A.: Wikify! Linking Documents to Encyclopedic Knowledge. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management. pp. 233–242. ACM, Lisbon, Portugal (2007)
13. Moro, A., Raganato, A., Navigli, R.: Entity linking meets word sense disambiguation: a unified approach. Transactions of the Association for Computational Linguistics (TACL) 2, 231–244 (2014)
14. Namee, P.M., Simpson, H., Dang, H.T.: Overview of the TAC 2009 Knowledge Base Population Track. In: Text Analysis Conference (TAC 2009) (2009)
15. Platt, J.C.: Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In: Advances in Kernel Methods - Support Vector Learning. MIT Press (January 1998)
16. Shen, W., Jianyong, W., Ping, L., Min, W.: LINDEN: Linking Named Entities with Knowledge Base via Semantic Knowledge. In: 21st International Conference on World Wide Web (WWW'12). pp. 449–458. Lyon, France (2012)
17. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: Issues, techniques, and solutions. Transactions on Knowledge and Data Engineering (2015)
18. Varma, V., Bharath, V., Kovelamudi, S., Bysani, P., GSK, S., N, K.K., Reddy, K., Kumar, K., Maganti, N.: IIT Hyderabad at TAC 2009. In: Text Analysis Conference (TAC 2009) (2009)
19. Zhang, W., Chuan, S.Y., Jian, S., Lim, T.C.: Entity Linking with Effective Acronym Expansion, Instance Selection and Topic Modeling. In: Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI-11). pp. 1909–1914. Barcelona, Catalonia, Spain (2011)
20. Zhang, W., Jian, S., Lim, T.C., Ting, W.W.: Entity Linking Leveraging: Automatically Generated Annotation. In: 23rd International Conference on Computational Linguistics (COLING 2010). pp. 1290–1298. Beijing, China (2010)
21. Zheng, Z., Xiance, S., Fangtao, L., Y, C.E., Xiaoyan, Z.: Entity Disambiguation with Freebase. In: 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT'12). pp. 82–89 (2012)