Collective disambiguation and Semantic Annotation for Entity Linking and Typing

Mohamed Chabchoub[1], Michel Gagnon[1], Amal Zouaq[2,1]

[1] École Polytechnique de Montral, Montral, Canada
{mohamed.chabchoub, michel.gagnon}@polymtl.ca
[2] University of Ottawa, Ottawa, Canada
azouaq@uottawa.ca

Abstract. In this paper we present our system for the 2016 OKE challenge Task 1. Our approach combines the output of a semantic annotator with the output of a named entity recognizer, and applies some heuristics for merging and filtering the detected mentions. The approach also applies a collective disambiguation method that relies on all the previously linked entities to choose between multiple candidate entities for a given mention. Using this approach, we greatly improve the performance of all the semantic annotators that are used as baselines in our experiments and also outperform the best system of the OKE Challenge 2015.

Keywords: OKE Challenge, Entity Recognition, Entity Typing, Entity linking

1 Introduction

The first task of the Open Knowledge Extraction (OKE) challenge is divided into three subtasks: entity recognition, entity linking and entity typing with four DUL classes (Person, Organization, Place and Role). In this paper, we show that by combining the outputs of a semantic annotator and a named entity recognizer, we can obtain very good performances for all three subtasks[3]. We rely on the principle of collective disambiguation [1, 5, 10], where the selection of the entity to be linked to a mention takes into account the entities previously associated to other mentions in the text. To implement our approach, we propose a pipeline architecture. In the first step, the detected mentions in text are adjusted and filtered, such that only the most relevant ones are transmitted to the entity linking module. The linking module identifies the corresponding entity in DBpedia for each mention and applies the collective disambiguation process if there are several candidates. Finally, for entity typing, our approach uses some manually defined mapping rules.

This paper is organized as follows. Next section discusses related works. Section 3 describes our system in more details. In Section 4 we present the results

[3] Our service is available at the following URL: http://westlab.polymtl.ca/OkeTask1/rest/annotate/post

of our system on the training dataset provided for the 2016 OKE Challenge and we compare it with several baselines, using different configurations. In Section 5 we conclude and make some suggestions to improve our system performance.

2 Related work

In this section we present some state-of-the-art approaches that perform well for named entity recognition and semantic annotation, with a particular focus on systems used in our evaluation for comparison purposes. We also present the best performing systems in the 2015 OKE challenge.

2.1 Named entity recognition

Extracting named entities has been tackled by numerous NLP researches in the last decades. OpenNLP[4] uses machine learning and maximum entropy models. LingPipe[5] uses n-gram character language models. OpenCalais[6] is a commercial service that uses machine learning techniques to recognize named entities and uses a proprietary taxonomy to type them. Stanford NER [6] is based on Conditional Random Fields (CRF) models and is widely used in the development of NLP applications. In an evaluation of named entity recognizers on bibliographical texts [2], Stanford NER was identified as the best system. Another evaluation on microposts [3] shows that Stanford NER is the second best after OpenCalais. This last result is especially relevant for our context, since we also have very short texts as input. Since OpenCalais is a commercial product, Stanford NER was thus our best choice. Note that this tool not only detects named entities mentions in a text, but also disambiguates them according to one of the following classes: PERSON, ORGANIZATION and LOCATION.

2.2 Semantic annotation

Most of the semantic annotators available as online services are commercial products. For our experiments, we decided to use non-commercial systems only.

DBpedia Spotlight [7] is a semantic annotator that uses a two-step process: first, entity mentions are spotted in text and then linked to some entity in DBpedia. The spotting phase relies on a set of surface forms extracted from DBpedia (titles, redirects) and anchors of Wikipedia links. To achieve entity linking, DBpedia Spotlight pre-ranks entity candidates for each surface form spotted in the text. It combines a prior score and a contextual score to determine which candidate entity is the most relevant. The prior score represents an estimation of how often the surface form is used as an anchor in a Wikipedia hyperlink that

[4] https://opennlp.apache.org/
[5] http://alias-i.com/lingpipe/
[6] http://www.opencalais.com/

points to the entity page. Formally it corresponds to the probability $P(e|s)$, where e and s are the entity and the surface form, respectively. The contextual score takes into account the context of the phrase (a window of words around the phrase) and the context of each candidate entity. Note that DBpedia has a confidence parameter that can be set before using it and whose value is between 0 and 1 (default value is 0.5). The effect of this parameter is to remove some annotations. The highest the confidence value, the highest is the probability of eliminating an annotation if it has more than one candidate for entity linking.

Similarly to Spotlight, Tagme [4] uses a list of surface forms extracted from Wikipedia anchors. For entity linking, it is based on a collective disambiguation process where, for each candidate entity of a surface form, a score of relatedness with the candidates of other surface forms is computed. The selected candidate is the one that maximizes a final score that combines all these relatedness scores. Tagme also uses a pruning method that retains only annotations that have a high link probability (defined as the number of Wikipedia articles that use it as an anchor, divided by the number of articles that mention it) or that have a high coherence score, which is computed by comparing it to all other annotations (by averaging over relatedness scores). Note that Tagme has been developed to be efficient with short texts, such as micro-blog posts.

Another semantic annotator that has been proposed recently is Babelfy [1], which uses a graph-based approach. Surface forms in text are associated to one or more vertices in BabelNet [8], a semantic network built from Wikipedia and Wordnet. Disambiguation is then achieved by a process that identifies the densest sub-graph according to some coherence metric.

Finally, AIDA [5] is a semantic annotator that assigns, to each spotted surface form, a value that corresponds to the prior score in DBpedia Spotlight, and a context similarity score. These scores are weighted using links and information extracted from DBpedia and YAGO. AIDA uses an additional score that estimates the coherence between two entities. This score is calculated using the Wikipedia inlinks of each entity. The main contribution of this system is a graph-based algorithm. The graph is composed of text mentions nodes and entities nodes. AIDA extracts the sub-graph which has the highest density, where the density is a combination of the three computed scores for each annotation.

2.3 Entity typing

None of the previous systems achieves the full task of the OKE Challenge, which includes entity typing according to some predefined types. Stanford NER does not consider the ROLE type. Semantic annotators only identify named entities in DBpedia, without typing them according to the OKE challenge type nomenclature. The best performing system for this task in the 2015 OKE Challenge [9] is ADEL [10], whose F-score is 0.60. It uses a hybrid approach that combines linguistic and semantic features. Linguistic resources, such as POS tagger, gazetteer and named entity recognizer, are used for the entity recognition task. For entity linking, some filtering is achieved, based on inbound and outbound links in Wikipedia. To select the most relevant entity among candidates, a graph-based approach similar to Babelfy is used. For typing with `dul:Role` class, ADEL mainly relies on a gazetteer (containing lists of occupations and nationalities). For other types, a manual alignment is built. Finally, a classifier is trained to filter out irrelevant entities. The second best-performing system, FOX-CETUS [11], has a F-score of about 0.5. Typing is achieved by using a pattern-based process that identifies the segment that expresses the type of the entity in the input. One important advantage of FOX-CETUS is that it is available through the GERBIL platform, thus facilitating peformance comparison.

3 System implementation

3.1 System Architecture

Our system is composed of three modules, as illustrated in Figure 1: 1) entity spotting, which consists of identifying relevant mentions in the text, 2) entity linking, where mentions are associated to some entity in DBpedia, when possible, and 3) entity typing, where the system tries to find, for each entity, the corresponding type in the DUL ontology. We will now describe each module.

3.2 Entity Spotting

The goal of this module is to extract all relevant mentions that can be found in the input text. We relied on semantic annotators freely available as web services and one named entity recognizer, the Stanford NER [6]. We experimented with four semantic annotators: DBpedia Spotlight, Babelfy, AIDA and Tagme. These semantic annotators not only detect relevant mentions in text, but also identify a corresponding entity in DBpedia. Stanford NER detects named entities that are tagged according to one of the following classes: *person*, *organization* and *location*. Each of these types can be aligned to a type in OKE challenge : *Person*, *Organization* and *Place*. However, the type *role*, which is also an OKE challenge type, is not recognized by the Stanford NER.

After running these semantic annotators and Stanford NER, we noticed a frequent overlap between mentions detected by semantic annotators and named entities extracted by Stanford NER. In this case, we keep the longest mention. Consider the following example, where two mentions found by DBpedia Spotlight are indicated in boldface:

> *John Stigall received a Bachelor of arts from the* ***State University*** *of* ***New York*** *at Cortland*

In this case Spotlight annotates *State University* and *New York* separately, whereas Stanford NER recognizes *State University of New York* as a single named entity. We select this last mention and discard the two separate mentions returned by Dbpedia Spotlight. This is done at the *Mention selection* step.

At this stage, we still do not always have the correct mention. In fact, in this example, the correct mention is *State University of New York at Cortland*.

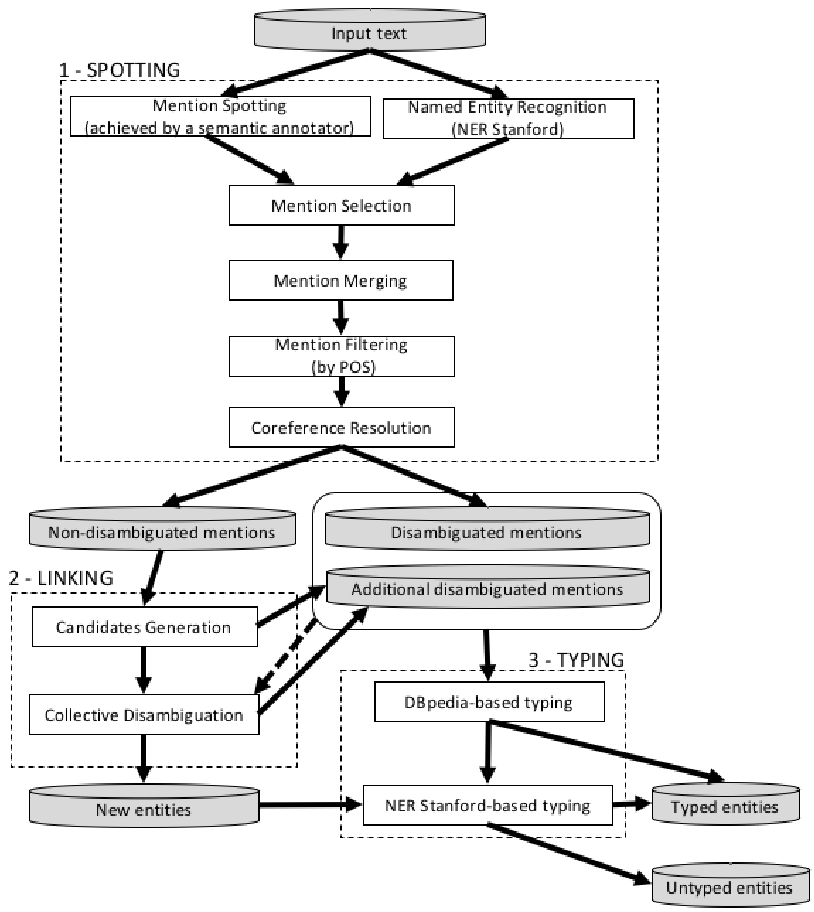

Fig. 1. System architecture

To deal with this problem, we developed a *mention merging algorithm*. Given a mention, the algorithm attempts to expand it to cover the next mention. Such expansion is permitted only if the second mention immediately follows the first one or if the mentions are separated by one of these patterns: a word, a comma, a comma followed by a word, a period or a period followed by a word. Following this rule, in the previous example the mention *State University of New York* will be expanded into *State University of New York at Cortland*. This process is repeated until we reach a state where the mention cannot be further expanded. The expanded mention obtained at each step is memorized, such that at the end we obtain a list of mentions that are ordered from the longest one to the shortest one. We then select the first mention in this list for which there is an entity in DBpedia whose label corresponds to this mention.

The next step is *mention filtering*, where the POS of each word in a mention is identified by using the Stanford POS tagger. Every mention that contains a verb is removed from the list. Finally, we use Stanford Coreference resolution to find, for each pronoun, the coreferent mention. The mention and its co-referents are thus linked to the same entity.

The output of this module consists in two sets. One set contains the mentions detected by the semantic annotator. These mentions are already disambiguated with their corresponding entities in DBpedia. The other set contains the named entities returned by Stanford NER and the new mentions that resulted from the merging algorithm. Remember that every mention that overlaps another mention is removed if the other one is longer. The second set does not contain any disambiguated mention and must be processed by the following module, which performs entity linking for these mentions. Note that mentions already detected by the semantic annotator are not added to this set.

3.3 Entity Linking

For every non-disambiguated mention, we query DBpedia to extract the entity whose label corresponds to the mention. If there is no such entity, we create a new URI resource as described in the OKE challenge. If there is one result, we check if it corresponds to a disambiguation page. If so, all entities that are referred by this disambiguation page are taken as candidates, and are given as input to the next step, that is, the disambiguation process. If the entity corresponds to a normal page (i.e not a disambiguation page), the entity is linked to the mention, and immediately added to the set of disambiguated entities.

When we have more than one candidate, the selection of the entity that will be linked to the mention is achieved by taking into account the other mentions that have already been disambiguated in the text. In this case, non-disambiguated mentions are processed sequentially. For each mention we compute a score for each candidate c associated to this mention. This score is based on the outlinks of the corresponding Wikipedia article of the entity c and is defined as follows:

$$Score(c) = \frac{\sum_{e_i \in G} outlink(e_i) \cap outlink(c)}{outlink(c) \cup outlink(G)}$$

where G is the set of all distinct entities that are already disambiguated.

We keep the candidate with the highest score value, and thus obtain a new linked mention that is added to the set of disambiguated entities. The process iterates until every mention has been disambiguated.

3.4 Entity Typing

The goal of this step is to align the extracted entities with one of the following classes in DUL ontology: DUL:Person, DUL:Organization, DUL:Place and DUL:Role. If a mention is not linked to a DBpedia entity, it is necessarily a mention that was recognized by Stanford NER. In this case, we simply apply the following mapping:

- Stanford:ORGANIZATION → DUL:Organization
- Stanford:LOCATION → DUL:Place
- Stanford:PERSON → DUL:Person

For mentions associated with a DBpedia URI, we try to find the type of the linked entity by executing the following SPARQL query:

```
SELECT ?type WHERE {<entity> rdf:type ?type }
```

If the query returns with success, the following mapping is applied, according to the instantiation of the variable `?type`:

- dbo:organisation → DUL:Organization
- yago:Organization108008335 → DUL:Organization
- dbo:Place → DUL:Place
- dbo:Location → DUL:Place
- dbo:EthnicGroup → DUL:Person
- dbo:Person → DUL:Person
- foaf:person → DUL:Person

If no type can be identified with this first query, we execute a second SPARQL query, where we check if the relevant information is represented by a predicate:

```
SELECT ?predicate WHERE {[] ?predicate <entity>}
```

According to the value extracted for the predicate, the following mapping is used:

- dbo:affiliation → DUL:Organization
- dbp:owner → DUL:Organization
- dbp:office → DUL:Organization
- dbo:birthPlace → DUL:Place
- dbo:location → DUL:Place
- dbo:deathPlace → DUL:Place
- dbo:occupation → DUL:Role
- dbp:occupation → DUL:Role

A conflict can appear with this method, when predicates are extracted with different types. This kind of conflict especially occurs between types *DUL:Role* and *DUL:Organization*, as for *State_University_of_New_York*, for which both predicates *dbo:affiliation* and *dbo:occupation* are extracted. In this case we count the number of occurrences of each predicate, and select the one with the highest number of occurrences. If no type can be identified with this second query, we execute a third SPARQL query, where we check if the relevant information is represented by a predicate, with the entity as subject:

```
SELECT ?predicate WHERE {<entity> ?predicate []}
```

According to the value extracted for the predicate, the following mapping is used:

– geo:geometry → DUL:Place

If this third query doesn't help to identity the type, we check if the mention was detected by Stanford NER in the spotting phase, and if so we use the mapping given at the beginning of this section. If not, the entity remains untyped.

4 Experiments and results

4.1 Training set description

We conducted our experiments against the training set given by the OKE challenge organizers, which is composed of 195 manually annotated sentences. It contains 1030 mentions annotated with 683 distinct entities, 530 among them are linked to DBpedia and 153 are defined as new entities. Table 1 provides detailed statistics on this dataset, for each type of entity.

Table 1. Statistics on the training set per DUL type

Type	# mentions	# disamb. mentions	# non-disamb. mentions
Dul:Role	165	144	21
Dul:Organization	237	198	39
Dul:Person	446	342	104
Dul:Place	182	171	11
Total	1030	855	175

Table 2. Results of the entity recognition, linking and typing tasks

	Entity Recognition			Entity Linking			Entity Typing			
	P	R	F	P	R	F	P	R	F	Average F
DBpedia Spot.	0.7388	**0.8430**	0.7875	0.5923	0.7224	0.6510	0.6538	**0.7498**	0.6985	0.7123
Babelfy	0.7742	0.8050	**0.7894**	0.6460	0.7091	0.6760	0.7031	0.7393	**0.7207**	**0.7287**
Tagme	0.7146	0.8153	0.7616	0.5912	**0.7317**	0.6540	0.6377	0.7315	0.6814	0.6990
AIDA	**0.8652**	0.6890	0.7671	**0.7670**	0.6273	**0.6901**	**0.7981**	0.6404	0.7106	0.7226

4.2 Results on the training set

To evaluate the potential of our approach, we applied it on the training set, using four publicly available semantic annotators: DBpedia Spotlight, Babelfy, Tagme and AIDA. We used the appropriate configuration of each system in order to have the highest number of possible mentions. For DBpedia Spotlight we set the confidence parameter at 0.0, while for Babelfy we used the default configuration, which does not limit the spotting phase to named entities only. We also used the default configuration for Tagme. For the purpose of our evaluation, we developed our own evaluation script following the OKE challenge criteria mentioned on the homepage of the competition. Only exact matches are counted. The macro-averaged precision-recall results are shown in Table 2. As we can see, the best performance was obtained by our framework coupled with Babelfy, in the three subtasks. However the difference with the F-scores obtained using the other semantic annotators is very low.

4.3 Baseline comparisons

We compared our system with the baselines of each semantic annotator used in our evaluation. To avoid improper comparison, we conducted two different evaluations. The first compares our system with Stanford NER (section 4.3.1) to evaluate its capabilities for the subtasks of entity recognition and typing. In the second evaluation (section 4.3.2), we compare our system with each of the employed semantic annotators to assess their initial performance in extracting and disambiguating named entities. We also compare the results of our system with the results of two systems that participated to the 2015 OKE Challenge. Our system is coupled with Babelfy semantic annotator, and achieves the best performance, according to Table 2.

4.3.1 Comparison with Stanford NER Since Stanford NER uses 3 classes only (Organization, Person and Place) we removed Role mentions from the training and results sets. The results are listed in Table 3. We can note that our system highly outperforms Stanford NER.

Table 3. Statistics on training set

	Entity Recognition			Entity Typing			
	P	R	F	P	R	F	Average F
Our system	0.8395	0.8183	0.8288	0.7923	0.7754	0.7837	**0.8063**
NER Stanford	0.8120	0.6740	0.7360	0.7360	0.6130	0.6690	0.6745

4.3.2 Comparison with semantic annotators We compared our system with the four semantic annotators used individually, under various configurations. For DBpedia Spotlight, we experimented with two confidence scores: 0.5 and 0.0. For Babelfy, we tested a configuration where only named entities are extracted. Finally, for Tagme, we also experimented a configuration that extracts mentions not linked to any entity (Tagme all extractions). The results are given in Table 4 and clearly show that our approach outperforms all the others.

Table 4. Comparison with semantic annotators

	Entity Recognition			Entity Linking			
	P	R	F	P	R	F	Average F
Our system	0.7792	0.8050	0.7894	0.6460	0.7091	0.6760	**0.7327**
Dbpedia Spotlight 0.0	0.3620	0.6370	0.4620	0.2650	0.5470	0.3570	0.4095
Dbpedia Spotlight 0.5	0.6570	0.4450	0.5310	0.5760	0.4630	0.5130	0.5220
AIDA	0.8245	0.5530	0.6620	0.7230	0.5160	0.6020	0.6320
Babelfy	0.3100	0.7340	0.4360	0.3250	0.6360	0.4330	0.4345
Babelfy NED	0.6500	0.4990	0.5640	0.6070	0.5470	0.5750	0.5695
Tagme	0.5110	0.7540	0.6090	0.4120	0.7060	0.5200	0.5645
Tagme (all extractions)	0.5080	0.7570	0.6080	0.4120	0.7060	0.5200	0.5640

Compared to semantic annotators, the increase in performance is at least 20 % for the F-score. Note that the improvement is more evident for the subtask of entity recognition.

4.3.3 Comparison with OKE Challenge 2015 To compare our approach to the participants of the 2015 OKE Challenge, we evaluated our system on the dataset of 2015 using GERBIL. This dataset is a subset of the Gold Standard provided for the 2016 OKE challenge. It contains 101 manually annotated sentences. For the comparison, we report the results of ADEL (the winner of OKE challenge 2015) that are found on the OKE 2105 Challenge homepage[7]. We also provide the performances of FOX-CETUS, the second best system, using GERBIL. We are thus evaluating the current version of the FOX-CETUS, whose performances are improved compared to the ones obtained in the 2015 Challenge. We see that our system obtains the best results.

Table 5. Statistics on evaluation dataset OKE 2015

	Micro F1	Micro P	Micro R	Macro F1	Macro P	Macro R
Our system	**0.7327**	**0.7638**	**0.7083**	**0.7216**	**0.7543**	**0.7076**
Adel	0.6075	0.6938	0.5403	0.6039	0.6850	0.5400
FOX	0.5378	0.6842	0.4528	0.5236	0.6670	0.4559

4.4 Analysis of errors

We analyzed some of the errors that our system generates. The most frequent errors are the following ones:

Spotting errors

As we explained in section 3.2, when there are overlapping mentions, we always keep the longest one. With this method, some errors might occur when the shortest mentions are in fact indicated in the gold standard. For example, we may obtain the mention *Paris, France*, whereas in the Gold Standard, we would

[7] https://github.com/anuzzolese/oke-challenge

have two separate mentions, one for *Paris* and another one for *France*. To avoid this kind of errors, some heuristics must be added to determine when a mention must be splitted.

Disambiguation errors

Some errors come from semantic annotators, since we directly take their annotations without any verification.

For the mentions that are disambiguated with our collective disambiguation algorithm, several reasons can explain the errors that occurred. Some errors may come from the candidates generator, which is based on an exact match of the label associated to the DBpedia entity. A partial match could be more appropriate but we must determine the threshold of similarity, and this will have a cost in terms of performance. Another source of error is the fact that our algorithm is based on the outlinks of each entity. Sometimes there are not enough mentions already disambiguated by the employed semantic annotator, and the number of outlinks is not sufficient to identify the right candidate.

Typing errors

In general, typing errors occur when a mention is incorrectly linked to an entity whose type is different from the expected one. Additionally, the DBpedia entities might be typed with more than one of the four types used in this task. For example, *National_Academy_of_Sciences* is typed as a *dbo:place* and a *yago:Organization108008335*. To avoid this kind of situation, we have to implement an algorithm that extracts the right type. Our current system simply gives priority to *dbo:Place*.

5 Conclusion

In this paper, we have shown that combining the outputs of a named entity recognizer and a semantic annotator, and applying a collective disambiguation approach for selecting among the candidate entities, results in a system that outperforms these two systems taken separately. We have also shown that by querying DBpedia to extract explicit and implicit types of entities, and using some manually defined mapping to the restricted set of four types used in the challenge, we also outperform the systems that participated to the 2015 OKE Challenge. Our approach is generic and does not depend on the dataset. In terms of efficiency, the most time-consuming steps are the queries to DBpedia, which are used for typing. For the identification of candidate URIs that correspond to some surface form, we use a local dump of DBpedia that has been indexed by Lucene.

For future work, we plan to improve the linking process and define a new approach to combine multiple annotators by taking advantage of their weaknesses and strengths.

6 Aknowledgement

This research has been funded by the NSERC Discovery Grant Program.

References

1. A. Raganato A. Moro and R. Navigli. Entity linking meets word sense disambiguation: a unified approach. In *TACL*.
2. Samet Atdag and Vincent Labatut. A comparison of named entity recognition tools applied to biographical texts. *CoRR*, abs/1308.0661, 2013.
3. Stefan Dlugolinsky, Marek Ciglan, and Michal Laclavik. Evaluation of named entity recognition tools on microposts. In *Intelligent Engineering Systems (INES), 2013 IEEE 17th International Conference on*, pages 197–202. IEEE, 2013.
4. P. Ferragina and U. Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *19th ACM conference on Information and knowledge management (CIKM)*.
5. I. Bordino H. Furstenau M. Pinkal M. Spaniol B. Taneva S. Thater J. Hoffart, M. A. Yosef and G. Weikum. Robust disambiguation of named entities in tex. In *8th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 782–792, Stroudsburg, PA, USA, 2011.
6. T. Grenager J. R. Finkel and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 363–370, 2005.
7. Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, pages 1–8, New York, NY, USA, 2011. ACM.
8. Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.
9. Andrea Giovanni Nuzzolese, Anna Lisa Gentile, Valentina Presutti, Aldo Gangemi, Darío Garigliotti, and Roberto Navigli. *Semantic Web Evaluation Challenges: Second SemWebEval Challenge at ESWC 2015, Portorož, Slovenia, May 31 - June 4, 2015, Revised Selected Papers*, chapter Open Knowledge Extraction Challenge, pages 3–15. Springer International Publishing, Cham, 2015.
10. Julien Plu, Giuseppe Rizzo, and Raphaël Troncy. *Semantic Web Evaluation Challenges: Second SemWebEval Challenge at ESWC 2015, Portorož, Slovenia, May 31 - June 4, 2015, Revised Selected Papers*, chapter A Hybrid Approach for Entity Recognition and Linking, pages 28–39. Springer International Publishing, Cham, 2015.
11. Michael Röder, Ricardo Usbeck, René Speck, and Axel-Cyrille Ngonga Ngomo. *Semantic Web Evaluation Challenges: Second SemWebEval Challenge at ESWC 2015, Portorož, Slovenia, May 31 - June 4, 2015, Revised Selected Papers*, chapter CETUS – A Baseline Approach to Type Extraction, pages 16–27. Springer International Publishing, Cham, 2015.