SemLinker, a Modular and Open Source Framework for Named Entity Discovery and Linking

Marie-Jean Meurs[*1], **Hayda Almeida**[2], **Ludovic Jean-Louis**[3], **Eric Charton**[4]
[1]Université du Québec à Montréal, [2]Concordia University, [3]Netmail, [4]Yellow Pages Group
Montreal, QC, Canada
meurs.marie-jean@uqam.ca, hayda.almeida@concordia.ca
ludovic.jean-louis@netmail.com, eric.charton@ypg.com
* corresponding author

Abstract

This paper presents SemLinker, an open source system that discovers named entities, connects them to a reference knowledge base, and clusters them semantically. SemLinker relies on several modules that perform surface form generation, mutual disambiguation, entity clustering, and make use of two annotation engines. SemLinker was evaluated in the English Entity Discovery and Linking track of the Text Analysis Conference on Knowledge Base Population, organized by the US National Institute of Standards and Technology. Along with the SemLinker source code, we release our annotation files containing the discovered named entities, their types, and position across processed documents.

Keywords: Entity Discovery, Entity Linking, SemLinker, TAC KBP

1. Introduction

Named Entity Recognition (NER) is the task of extracting and classifying Named Entities (NEs) in textual documents, while Entity Linking (EL) consists in connecting NEs to their corresponding entries in a Knowledge Base (KB) such as Wikipedia. The English Entity Discovery and Linking (EDL) task (Ji et al., 2014) at the Text Analysis Conference on Knowledge Base Population (TAC KBP[1]) consists in performing full NE discovery, as well as linking and clustering discovered NEs. To handle this task, automated systems identify NEs in a given document collection, and link them to their corresponding entities in a reference KB if they exist. Discovered NEs are of type person (PER), organization (ORG), and geo-political entity (GPE). For example, in a document with the following content: *“Obama visited Cambridge”*, NEs are annotated as “$\llcorner \mathit{Obama} \lrcorner_{[PER]}$ *visited* $\llcorner \mathit{Cambridge} \lrcorner_{[ORG]}$”, and linked to relevant Wikipedia entries `[Barack_Obama]` and `[University_of_Cambridge]`. NEs linked to KB entries are naturally clustered together, while entities that are not present in the KB, called NIL entities, have to be semantically clustered. Extracting NEs in a document collection and associating them to a KB is a complex task. Relevant entities are often polysemous, and can thus be associated with several nodes in a KB. For instance, a linking system would find multiple candidate nodes in a Wikipedia-based KB to link *“Cambridge”* to, since the Wikipedia `Cambridge_(disambiguation)` page contains no less than 54 entries. EDL systems have hence to be able to disambiguate NEs.

The EDL task is currently investigated in the natural language processing community with various goals such as populating a new KB from a raw document collection, or improving the retrieval performance of search engines. The SemLinker EDL system was first presented in the TAC KBP Entity Linking 2013 task (Charton et al., 2013; Charton et al., 2014a). In 2014, the scope of the task was enlarged to also cover NE discovery, and SemLinker evolved according to this new requirement. The current version of the system relies on generic annotation engines to discover and link NEs in documents. A pipeline approach is used: First, mutual relations within documents are considered (Charton et al., 2014b) for candidate NEs to improve the entity linking accuracy. Second, the results provided by different annotation engines are merged in a collaborative approach. To perform this merge, SemLinker makes use of two annotation engines, Wikimeta (Charton and Gagnon, 2012), and AIDA (Hoffart et al., 2011b). The engines provide Wikipedia or DBPedia links for each annotated NE found in the document collection. Different disambiguation processes and rule-based merging strategies are considered according to the annotation engine. These strategies are based on heuristics derived from the performance of the annotation engines observed on the training data.

This paper is organized as follows: Section 2. describes some related works while Section 3. provides details about the system resources. The proposed algorithms along with the system architecture are presented in Section 4. Experiments and results are reported in Section 5., and we conclude in Section 6.

2. Related work

The EDL task (Ji et al., 2014) requires a system to execute NER, disambiguation and linking. Several studies address the problem of performing full named entity extraction using Wikipedia (Cucerzan, 2007; Milne and Witten, 2008), a specific KB such as YAGO2 (Hoffart et al., 2011b), or Freebase (Sil and Yates, 2013).

[1] http://www.nist.gov/tac/2014/index.html

Approaches that combine knowledge from multiple resources have already been explored in similar research problems. In the field of information retrieval (IR), a commonly used merging technique is called data fusion. This method attempts to combine the output of multiple IR systems to improve the quality of a retrieved list of documents with regards to a search query (Nuray and Can, 2006). Such fused strategies were shown to usually outperform regular results (Wu and McClean, 2006).

Previous works have also used combined approaches to improve NE annotation. Finkel and Manning describe a strategy that applies knowledge acquired from sentence parsing to help improve the entity discovery process (Finkel and Manning, 2009). Another combined approach was presented by Chen and Ji (2011) to perform EL ranking. However, to the best of our knowledge, a model that implements a combination between entity annotation engines has not yet been described.

3. System Resources

SemLinker utilizes two corpus-based resources. The first one is a Wikipedia dump that was downloaded in July 2013, and indexed with Lucene-search for Wiki. The Wikipedia dump is used in the system as an internal link and category resource for the mutual disambiguation algorithm. The second corpus resource is NLGbAse[2] (Charton and Torres-Moreno, 2010). NLGbAse is a multilingual linguistic resource that provides a set of metadata for each document found in Wikipedia. The process to build NLGbAse metadata is described in (Bunescu and Pasca, 2006), and this approach is applied in (Charton and Torres-Moreno, 2010).

The system also makes use of two annotation engines: Wikimeta (Charton and Gagnon, 2012), and AIDA[3], the open source named entity disambiguation system described in (Hoffart et al., 2011b). Wikimeta provides various layers of annotations for a given document, and its components have been described in (Charton and Gagnon, 2012). Wikimeta is able to annotate over 3 million entities, which represent more than the 800k entities that were present in the KB. This enables the system to correctly identify many NEs considered as NILs in the TAC KBP evaluation framework. The use of NLGbAse to support the disambiguation algorithm improves the SemLinker clustering module effectiveness on NIL entities.

AIDA is an open source framework for entity detection and disambiguation, described in (Hoffart et al., 2011b). AIDA is capable of mapping mentions of ambiguous NEs onto canonical entities (persons, organizations, locations...) found in the Wikipedia-derived YAGO2 knowledge base (Hoffart et al., 2011a). This annotation engine proposes a disambiguation method that combines popularity-based priors, similarity measures, coherence, and semantic relatedness.

4. System Components

SemLinker is an open source software. Its components are developed in a modular way, making them easily reusable. The system pipeline uses the following modules to process documents.

For each annotation engine:

- **NE Discovery**: finding NEs in each document;
- **NE Annotation**: assigning each NE with PER, ORG, GPE labels, Part Of Speech (POS) tags, and a ranked set of candidate Uniform Resource Identifiers (URIs) from Wikipedia;
- **Correction Processes**: using annotation layers to correct NE labels and first ranked URIs accross a co-reference chain;
- **Mutual Disambiguation**: re-ranking of candidate URIs for each NE;
- **Extraction of the best link** for each NE.

Once all the documents have been processed by both annotation engines:

- **Annotation combination**: merging NE lists and linking results from different annotation engines;
- **NIL clustering**: grouping together similar NEs not found in the KB.

A brief description of these modules is given hereafter.

4.1. Entity Extraction and Annotation

Annotation engines provide relevant knowledge for the documents in the collection by extracting all NEs found, and assigning them a POS tag, a surface form, a NE label, and a list of ranked candidate Wikipedia URIs. The NE data found in each document is used to generate a matrix representation of the document called the annotation object. In addition to all the information provided by annotation engines for a given NE, the system also keeps the offset spans for each word in the annotation object.

Two correction processes are applied to all NEs found to be associated with the same first ranked URI. The processes are based on co-reference chains and NE label frequency. To identify co-reference chains, the system relies on the information stored in the annotation object. NEs within the same document are clustered into co-reference chains, and the same URI is assigned to all the NEs belonging to a chain. The decision process is based on weighting the most frequent URI and the longest NE surface form in a chain.

The next step is the normalization of NE labels in the same co-reference chain. Labels of NEs that share the same common first ranked URI are updated with the most frequent NE label found in the co-reference chain. The decision process takes into account the NE label occurrences in each cluster of identical first ranked URIs. The most frequent NE label in a cluster is assigned to all the NEs in the cluster.

To improve annotation precision, URIs assigned to NEs are re-ranked based on a mutual disambiguation approach, as described in (Charton et al., 2014b).

To perform mutual disambiguation, the algorithm uses the information found in the annotation object to generate a

[2] http://www.nlgbase.org

[3] https://github.com/yago-naga/aida

NE Label Doc type	PER	ORG	GPE
Discussion Forum (DF)	NIL: **W** KB: **W**	NIL: **A** KB: **W**	NIL: **A** KB: **A**
Web data (WB)	NIL: **W** KB: **W**	NIL: **A** KB: **A**	NIL: **A** KB: **W**
Newswire (NW)	NIL: **W** KB: **W**	NIL: **A** KB: **A**	NIL: **W** KB: **W**

Table 1: Best annotator on NE discovery between Wikimeta (**W**) and AIDA (**A**) on the training dataset.

	Training set	Test set
Documents	158	138
Queries	5,966	5,234
PER Label	3,193 (53.5%)	3,162 (60.4%)
ORG Label	1,340 (22.5%)	1,007 (19.2%)
GPE Label	1,433 (24.0%)	1,065 (20.4%)
NIL node	2,624 (44.0%)	2,417 (46.2%)
KB node	3,342 (56.0%)	2,817 (53.8%)
DF type	2,029 (34.0%)	1,916 (36.6%)
NW type	2,773 (46.5%)	1,575 (30.1%)
WB type	1,164 (19.5%)	1,743 (33.3%)

Table 2: Training and test dataset statistics.

graph containing all the internal links and categories encountered in the Wikipedia source document related to each candidate URI assigned to a NE. The algorithm computes a mutual relation score for each candidate URI of each NE. The score of a URI candidate is compared to the scores of all other candidate URIs of NEs found in the document. Candidate URIs are then re-ordered by decreasing order of mutual relation score.

4.2. Annotation Combination

SemLinker utilizes annotations from Wikimeta and AIDA engines, described in Section 3., to provide a combined approach to the EDL task. Output combination strategies were previously used in natural language processing tasks to enhance overall results, as described in (Zhang et al., 2009; Po and Bergamaschi, 2010). We developed and evaluated several heuristics for combining the outcome from both annotators to improve wikification results. The global strategy is described hereafter, while Subsection 5.3. provides more details about the heuristics developed for this approach.

Combining the output from annotators is required since some NEs have been discovered by both annotation engines, while others come from only one annotator. NEs annotated by both engines are considered for merge only if they share the same offset span in the original document. When identified by two engines, a given NE can potentially be associated to different NE labels and URIs. The information associated to each NE is used to select the best annotation candidate. The combination algorithm evaluates NE labels and URIs provided by each engine, as well as the document type from which the NE was extracted. Document types found in the corpora are news documents (NW), blog posts (WB), and forum contributions (DF). If a NE is discovered by only one of the annotation engines, the combination algorithm also takes into account the performance achieved by the annotation engine with the TAC KBP training data.

Table 1 shows the best annotator on NE discovery between Wikimeta and AIDA on the training dataset. The decision rules used to develop the heuristics for the combination process were based on these results. The rules were defined based on which engine performed better in the discovery of different NEs labels, document type or NIL/KB status. The flexibility of the algorithm allows fine-grained merging strategies to benefit from the best capabilities of each annotation engine in terms of labeling, linking and dealing with complex documents.

After combination of the ouputs, NE clustering is performed in two steps. First, NEs are clustered according to their assigned URIs if available. NEs that are associated to the same KB node are naturally clustered together, as well as NIL NEs that were assigned identical URIs. Then, for all NIL NEs that were not assigned a URI, new NIL clusters are created. The clustering algorithm applies a substring strategy to add NIL NEs to existing clusters, or creates new clusters for NIL NEs with similar surface forms. Finally, orphan NEs are clustered as singletons.

5. Experiments and Results

This Section presents the data utilized to run the EDL experiments, a description of the metrics applied on the task, and the results obtained by SemLinker at TAC KBP.

5.1. Training and Test Corpora

The training set used by SemLinker 2014 was the TAC KBP 2014 English EDL training data (LDC2014E54). The test set was the TAC KBP 2014 English EDL evaluation source corpus (LDC2014E87). Documents in the datasets come from Web data (WB), Newswire (NW), and Discussion Fora (DF).

The training set contains a total of 158 documents and 5,966 NEs, while the test set has 138 documents and

System	DiscP	DiscR	DiscF	LinkP	LinkR	LinkF	CEAFmP	CEAFmR	CEAFmF
Wikimeta only	0.546	0.595	0.570	0.503	0.548	0.524	0.561	0.611	0.585
AIDA only	0.647	0.471	0.545	0.508	0.370	0.428	0.626	0.456	0.528
heuristic 1	0.533	0.638	0.581	0.487	0.583	0.531	0.541	0.647	0.589
heuristic 2	0.543	0.604	0.572	0.500	0.556	0.527	0.557	0.619	0.586
heuristic 3	0.539	0.620	0.576	0.494	0.569	0.529	0.551	0.634	0.589
heuristic 4	0.606	0.418	0.495	0.562	0.388	0.459	0.624	0.430	0.510

Table 3: SemLinker performance on the training corpus according to annotation engine or heuristic applied.

System	DiscP	DiscR	DiscF	LinkP	LinkR	LinkF	CEAFmP	CEAFmR	CEAFmF
Wikimeta only	0.549	0.579	0.563	0.498	0.525	0.511	0.542	0.572	0.557
heuristic 1	0.539	0.648	0.589	0.481	0.577	0.525	0.517	0.621	0.564
heuristic 2	0.548	0.589	0.568	0.496	0.533	0.514	0.539	0.580	0.559
heuristic 3	0.547	0.616	0.579	0.492	0.554	0.521	0.533	0.600	0.565
heuristic 4	0.632	0.416	0.502	0.565	0.372	0.449	0.619	0.408	0.492

Table 4: SemLinker performance on the test corpus according to annotation engine or heuristic applied.

5,234 NEs. Table 2 provides more details about the training and test corpora utilized in our experiments. It is interesting to notice how the different categories of NEs are not well-balanced among the corpora: more than half of the NEs are of type Person. One can also observe that only a little more than 50% of all NEs are associated with a KB node.

5.2. Metrics

The TAC KBP EDL task is split in three different subtasks: discovery, linking, and clustering. SemLinker is evaluated on these sub-tasks using different metrics: Disc (discovery), Link (linking), and CEAFm (CEAF (Luo, 2005)) Precision, Recall and F-measure.

Clustering performance is measured by the CEAF score, which computes the similarity between the gold-standard and the system output. Name tagging is evaluated by the Disc score, which verifies the correctness of entity name boundaries and types. The linking performance is measured by the Link score, which verifies the performance on NEs that were assigned KB nodes. More detailed definitions of the TAC KBP metrics are provided in the TAC 2014 workshop notebook (Ji et al., 2014).

5.3. Combination Strategies

To merge results from both annotation engines, the combination algorithm selects or rejects NEs and candidate links depending on the NE entity label (PER, ORG, GPE), the document type (DF, NW, WB), and the NE linking status (KB or NIL). In total, 12 heuristics were developed and evaluated. We describe the ones presenting the best performance in the training data: $h1$, $h2$, $h3$, and $h4$. The motivation to use heuristic $h4$ was that it obtained the best performance for precision results, while $h1$ to $h3$ obtained the best F-measure scores on the training set.

The first heuristic ($h1$) adopts a "more-is-better" approach. Candidates triples (NE, NE label, link)$_W$ obtained from Wikimeta are kept as is, while those from AIDA are added only if they contain NEs not found among Wikimeta candidates. In $h1$ priority is given to Wikimeta candidates because this engine outperformed AIDA in overall performance on the EDL 2014 training set.

The second heuristic ($h2$) selects Wikimeta candidates, and adds AIDA candidates that contain an ORG NE label and that are not found among Wikimeta candidates.

The third heuristic ($h3$) is similar to $h2$, but in $h3$ AIDA candidates are added only if they present an ORG NE label and if the candidate is found in the KB (it is not a NIL).

Finally, the last heuristic ($h4$) keeps all the Wikimeta candidates, except candidates assigned an ORG NE label, and adds all AIDA candidates with an ORG NE label.

5.4. Results

The results obtained by SemLinker on the training data are presented in Table 3. Runs with Wikimeta only and AIDA only are obtained from the annotators, without any merging strategy. The detailed results obtained on the test set are presented in Table 4.

The results obtained on the test set are consistent with the performance of the annotation engines in the training data. Heuristics $h1$ and $h2$ show that selecting more Wikimeta candidates increases the recall for all metrics. When priority is given to AIDA candidates, as implemented by $h4$, the precision improves. The results obtained by $h3$ and $h1$ combinations outperformed the results obtained either with Wikimeta candidates only or with AIDA candidates only. This well-balanced combination approach allows SemLinker to benefit from the strength of both annotation engines.

6. Discussion and Conclusion

The combination approach demonstrates better performance compared to the results obtained only by a single annotation engine. The best results for entity discovery and linking were presented by heuristic 1. This strategy is oriented towards a high recall for NE discovery and linking, which decreases the precision results. The best precision results were obtained by heuristic 4, and a balanced strategy that demonstrates interesting results for both precision and recall overall is heuristic 3. This strategy enriches results from Wikimeta, while not introducing much noise in the entities list.

Reproducibility and data availability.
As was SemLinker 2013, SemLinker 2014 is publicly released as an open source software available in the following repository:
https://github.com/SemLinker-Team/SemLinker_KBP2014
We also release our annotation files for the TAC KBP EL 2013 and EDL 2014 training and test sets.

7. Bibliographical References

Bunescu, R. C. and Pasca, M. (2006). Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, volume 6. Association for Computational Linguistics.

Charton, E. and Gagnon, M. (2012). A Disambiguation Resource Extracted from Wikipedia for Semantic Annotation. In *Proceedings of LREC 2012*.

Charton, E. and Torres-Moreno, J. (2010). NLGbAse: A Free Linguistic Resource for Natural Language Processing Systems. In *Proceedings of LREC 2010*.

Charton, E., Meurs, M.-J., Jean-Louis, L., and Gagnon, M. (2013). SemLinker System for KBP2013: A Disambiguation Algorithm Based on Mutual Relations of Semantic Annotations Inside a Document. In *Proceedings of the Text Analysis Conference 2013 (TAC2013)*.

Charton, E., Meurs, M.-J., Jean-Louis, L., and Gagnon, M. (2014a). Improving Entity Linking Using Surface Form Refinement. In *Proceedings of LREC 2014*.

Charton, E., Meurs, M.-J., Jean-Louis, L., and Gagnon, M. (2014b). Mutual Disambiguation for Entity Linking. In *Proceedings of ACL 2014, The 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, June 22-27, 2014*.

Cucerzan, S. (2007). Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing - CoNLL*, volume 7, pages 708–716. Association for Computational Linguistics.

Finkel, J. R. and Manning, C. D. (2009). Joint Parsing and Named Entity Recognition. In *Proceedings of the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 326–334. Association for Computational Linguistics.

Hoffart, J., Suchanek, F. M., Berberich, K., Lewis-Kelham, E., De Melo, G., and Weikum, G. (2011a). Yago2: Exploring and Querying World Knowledge in Time, Space, Context, and many Languages. In *Proceedings of the 20th International Conference Companion on World Wide Web*, pages 229–232. ACM.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. (2011b). Robust Disambiguation of Named Entities in Text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics.

Ji, H., Dang, H. T., Nothman, J., and Hachey, B. (2014). Overview of TAC-KBP 2014 Entity Discovery and Linking Tasks. In *Proceedings of the Text Analysis Conference (TAC2014)*.

Luo, X. (2005). On Coreference Resolution Performance Metrics. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 25–32. Association for Computational Linguistics.

Milne, D. and Witten, I. H. (2008). Learning to Link with Wikipedia. In *Proceedings of the 17th ACM International Conference on Information and Knowledge Management*, CIKM '08, pages 509–518. ACM.

Nuray, R. and Can, F. (2006). Automatic Ranking of Information Retrieval Systems Using Data Fusion. *Information Processing & Management*, 42(3):595–614.

Po, L. and Bergamaschi, S. (2010). Automatic Lexical Annotation Applied to the SCARLET Ontology Matcher. In *Intelligent Information and Database Systems*, pages 144–153. Springer.

Sil, A. and Yates, A. (2013). Re-ranking for Joint Named-entity Recognition and Linking. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, CIKM '13, pages 2369–2374. ACM.

Wu, S. and McClean, S. (2006). Performance Prediction of Data Fusion for Information Retrieval. *Information Processing & Management*, 42(4):899–915.

Zhang, H., Zhang, M., Tan, C. L., and Li, H. (2009). K-best Combination of Syntactic Parsers. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3*, pages 1552–1560. Association for Computational Linguistics.