Mutual Disambiguation for Entity Linking

Eric Charton
Polytechnique Montréal
Montréal, QC, Canada
eric.charton@polymtl.ca

Marie-Jean Meurs
Concordia University
Montréal, QC, Canada
marie-jean.meurs@concordia.ca

Ludovic Jean-Louis
Polytechnique Montréal
ludovic.jean-louis@polymtl.ca

Michel Gagnon
Polytechnique Montréal
michel.gagnon@polymtl.ca

Abstract

The disambiguation algorithm presented in this paper is implemented in SemLinker, an entity linking system. First, named entities are linked to candidate Wikipedia pages by a generic annotation engine. Then, the algorithm re-ranks candidate links according to mutual relations between all the named entities found in the document. The evaluation is based on experiments conducted on the test corpus of the TAC-KBP 2012 entity linking task.

1 Introduction

The Entity Linking (EL) task consists in linking name *mentions* of named entities (NEs) found in a document to their corresponding entities in a reference Knowledge Base (KB). These NEs can be of type person (PER), organization (ORG), etc., and they are usually represented in the KB by a Uniform Resource Identifier (URI). Dealing with ambiguity is one of the key difficulties in this task, since mentions are often highly polysemous, and potentially related to many different KB entries. Various approaches have been proposed to solve the named entity disambiguation (NED) problem. Most of them involve the use of surface forms extracted from Wikipedia. Surface forms consist of a word or a group of words that match lexical units like *Paris* or *New York City*. They are used as matching sequences to locate corresponding candidate entries in the KB, and then to disambiguate those candidates using similarity measures.

The NED problem is related to the *Word Sense Disambiguation (WSD)* problem (Navigli, 2009), and is often more challenging since mentions of NEs can be highly ambiguous. For instance, names of places can be very common as is Paris, which refers to 26 different places in Wikipedia. Hence, systems that attempt to address the NED problem must include disambiguation resources. In the context of the Named Entity Recognition (NER) task, such resources can be generic and generative. This generative approach does not apply to the EL task where each entity to be linked to a semantic description has a specific word context, marker of its exact identity.

One of the classical approach to conduct the disambiguation process in NED applications is to consider the context of the mention to be mapped, and compare this context with contextual information about the potential target entities (see for instance the KIM system (Popov et al., 2003)). This is usually done using similarity measures (such as cosine similarity, weighted Jaccard distance, KL divergence...) that evaluate the distance between a bag of words related to a candidate annotation, and the words surrounding the entity to annotate in the text.

In more recent approaches, it is suggested that annotation processes based on similarity distance measures can be improved by making use of other annotations present in the same document. Such techniques are referred to as *semantic relatedness* (Strube and Ponzetto, 2006), *collective disambiguation* (Hoffart et al., 2011b), or *joint disambiguation* (Fahrni et al., 2012). The idea is to evaluate in a set of candidate links which one is the most likely to be correct by taking the other links contained in the document into account. For example, if a NE describes a city name like *Paris*, it is more probable that the correct link for this city name designates *Paris (France)* rather than *Paris (Texas)* if a neighbor entity offers candidate links semantically related to *Paris (France)* like the *Seine river* or the *Champs-Elysées*. Such techniques mostly involve exploration of graphs resulting of all the candidate annotations proposed for a given document, and try to rank the best candidates for each annotation using an ontology. The ontology (like YAGO or DBPedia) provides a pre-

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 476–481,
Baltimore, Maryland, USA, June 23-25 2014.

existing set of potential relations between the entities to link (like for instance, in our previous example, *Paris (France)* `has_river` *Seine*) that will be used to rank the best candidates according to their mutual presence in the document.

In this paper we explore the capabilities of a disambiguation algorithm using all the available annotation layers of NEs to improve their links. The paper makes the following novel propositions: 1) the ontology used to evaluate the relatedness of candidates is replaced by internal links and categories from the Wikipedia corpus; 2) the coherence of entities is improved prior to the calculation of semantic relatedness using a co-reference resolution algorithm, and a NE label correction method; 3) the proposed method is robust enough to improve the performance of existing entity linking annotation engines, which are capable of providing a set of ranked candidates for each annotation in a document.

This paper is organized as follows. Section 2 describes related works. The proposed method is presented in Section 3 where we explain how our SemLinker system prepares documents that contain mentions to disambiguate, then we detail the disambiguation algorithm. The evaluation of the complete system is provided in Section 4. Finally, we discuss the obtained results, and conclude.

2 Related Work

Entity annotation and linking in natural language text has been extensively studied in NLP research. A strong effort has been conducted recently by the TAC-KBP evaluation task (Ji et al., 2010) to create standardized corpus, and annotation standards based on Wikipedia for evaluation and comparison of EL systems. In this paper, we consider the TAC-KBP framework. We describe below some recent approaches proposed for solving the EL task.

2.1 Wikipedia-based Disambiguation Methods

The use of Wikipedia for explicit disambiguation dates back to (Bunescu and Pasca, 2006) who built a system that compared the context of a mention to the Wikipedia categories of an entity candidate. Lately, (Cucerzan, 2007; Milne and Witten, 2008; Nguyen and Cao, 2008) extended this framework by using richer features for similarity comparison. Some authors like Milne and Witten (2008) utilized machine learning methods rather than a similarity function to map mentions to entities. They also introduced the notion of semantic relatedness. Alternative propositions were suggested in other works like (Han and Zhao, 2009) that considered the relatedness of common noun phrases in a mention context with Wikipedia article names. While all these approaches focus on semantic relation between entities, their potential is limited by the separate mapping of candidate links for each mention.

2.2 Semantic Web Compliant Methods

More recently, several systems have been launched as web services dedicated to EL tasks. Most of them are compliant with new emergent semantic web standards like LinkedData network. DBPedia Spotlight (Mendes et al., 2011) is a system that finds mentions of DBpedia (Auer et al., 2007) resources in a textual document. Wikimeta (Charton and Gagnon, 2012) is another system relying on DBpedia. It uses bags of words to disambiguate semantic entities according to a cosine similarity algorithm. Those systems have been compared with commercial ones like AlchemyAPI, Zemanta, or Open Calais in (Gangemi, 2013). The study showed that they perform differently on various essential aspects of EL tasks (mention detection, linking, disambiguation). This suggests a wide range of potential improvements on many aspects of the EL task. Only some of these systems introduce the semantic relatedness in their methods like the AIDA (Hoffart et al., 2011b) system. It proposes a disambiguation method that combines popularity-based priors, similarity measures, and coherence. It relies on the Wikipedia-derived YAGO2 (Hoffart et al., 2011a) knowledge base.

3 Proposed Algorithm

We propose a mutual disambiguation algorithm that improves the accuracy of entity links in a document by using successive corrections applied to an *annotation object* representing this document. The annotation object is composed of information extracted from the document along with linguistic and semantic annotations as described hereafter.

3.1 Annotation Object

Documents are processed by an annotator capable of producing POS tags for each word, as well as spans, NE surface forms, NE labels and ranked candidate Wikipedia URIs for each candidate NE. For each document $\mathcal{D}$, this knowledge is gathered

in an array called *annotation object*, which has initially one row per document lexical unit. Since the system focuses on NEs, rows with lexical units that do not belong to a NE SF are dropped from the annotation object, and NE SF are refined as described in (Charton et al., 2014). When NE SF are spanned over several rows, these rows are merged into a single one. Thus, we consider an annotation object $\mathcal{A}_\mathcal{D}$, which is an array with a row for each NE, and columns storing related knowledge.

If n NEs were annotated in $\mathcal{D}$, then $\mathcal{A}_\mathcal{D}$ has n rows. If l candidate URIs are provided for each NE, then $\mathcal{A}_\mathcal{D}$ has $(l + 4)$ columns $c_{u,u\in\{1,l+4\}}$. Columns c_1 to c_l store Wikipedia URIs associated with NEs, ordered by decreasing values of likelihood. Column c_{l+1} stores the offset of the NEs, c_{l+2} stores their surface forms, c_{l+3} stores the NE labels (PER, ORG, ...), and c_{l+4} stores the (vectors of) POS tags associated with the NE surface forms. $\mathcal{A}_\mathcal{D}$ contains all the available knowledge about the NEs found in $\mathcal{D}$. Before being processed by the disambiguation module, $\mathcal{A}_\mathcal{D}$ is dynamically updated by correction processes.

3.2 Named Entity Label Correction

To support the correction process based on co-reference chains, the system tries to correct NE labels for all the NEs listed in the *annotation object*. The NE label correction process assigns the same NE label to all the NEs associated with the same first rank URI. For all the rows in $\mathcal{A}_\mathcal{D}$, sets of rows with identical first rank URIs are considered. Then, for each set, NE labels are counted per type, and all the rows in a same set are updated with the most frequent NE label found in the set, i.e. all the NEs in this set are tagged with this label.

3.3 Correction Based on Co-reference Chains

First rank candidate URIs are corrected by a process that relies on co-reference chains found in the document. The co-reference detection is conducted using the information recorded in the annotation object. Among the NEs present in the document, the ones that co-refer are identified and clustered by logical rules applied to the content of the annotation object. When a co-reference chain of NEs is detected, the system assigns the same URI to all the members of the chain. This URI is selected through a decision process that gives more weight to longer surface forms and frequent URIs. The following example illustrates an application of this correction process:

Three sentences are extracted from a document about Paris, the French capital. NEs are indicated in brackets, first rank URIs and surface forms are added below the content of each sentence.
- [Paris] is famous around the world.
URI$_1$: `http://en.wikipedia.org/wiki/Paris_Hilton`
NE surface form: `Paris`
- The [city of Paris] attracts millions of tourists.
URI$_1$: `http://en.wikipedia.org/wiki/Paris`
NE surface form: `city of Paris`
- The [capital of France] is easy to reach by train.
URI$_1$: `http://en.wikipedia.org/wiki/Paris`
NE surface form: `capital of France`

The three NEs found in these sentences compose a co-reference chain. The second NE has a longer surface form than the first one, and its associated first rank URI is the most frequent. Hence, the co-reference correction process will assign the right URI to the first NE (URI$_1$: `http://en.wikipedia.org/wiki/Paris`), which was wrongly linked to the actress Paris Hilton.

3.4 Mutual Disambiguation Process

The extraction of an accurate link is a process occurring after the URI annotation of NEs in the whole document. The system makes use of all the semantic content stored in $\mathcal{A}_\mathcal{D}$ to locally improve the precision of each URI annotation in the document. The Mutual Disambiguation Process (MDP) relies on the graph of all the relations (internal links, categories) between Wikipedia content related to the document annotations.

A basic example of semantic relatedness that should be captured is explained hereafter. Let us consider the mention *IBM* in a given document. Candidate NE annotations for this mention can be *International Business Machine* or *International Brotherhood of Magicians*. But if the *IBM* mention co-occurs with a *Thomas Watson, Jr* mention in the document, there will probably be more links between the *International Business Machine* and *Thomas Watson, Jr* related Wikipedia pages than between the *International Brotherhood of Magicians* and *Thomas Watson, Jr* related Wikipedia pages. The purpose of the MDP is to capture this semantic relatedness information contained in the graph of links extracted from Wikipedia pages related to each candidate annotation.

In MDP, for each Wikipedia URI candidate annotation, all the internal links and categories contained in the source Wikipedia document related

to this URI are collected. This information will be used to calculate a weight for each of the l candidate URI annotations of each mention. For a given NE, this weight is expected to measure the mutual relations of a candidate annotation with all the other candidate annotations of NEs in the document. The input of the MDP is an annotation object $\mathcal{A}_\mathcal{D}$ with n rows, obtained as explained in Section 3.1. For all $i \in [\![1, n]\!]$, $k \in [\![1, l]\!]$, we build the set S_i^k, composed of the Wikipedia URIs and categories contained in the source Wikipedia document related to the URI stored in $\mathcal{A}_\mathcal{D}[i][k]$ that we will refer to as URI_i^k to ease the reading.

Scoring:

For all $i,\ j \in [\![1, n]\!]$, $k \in [\![1, l]\!]$, we want to calculate the weight of mutual relations between the candidate URI_i^k and all the first rank candidates URI_j^1 for $j \neq i$. The calculation combines two scores that we called *direct semantic relation* score (dsr_score) and *common semantic relation* score (csr_score):

- the dsr_score for URI_i^k sums up the number of occurrences of URI_i^k in S_j^1 for all $j \in [\![1, n]\!] - \{i\}$.

- the csr_score for URI_i^k sums up the number of common URIs and categories between S_i^k and S_j^1 for all $j \in [\![1, n]\!] - \{i\}$.

We assumed the dsr_score was much more semantically significant than the csr_score, and translated this assumption in the weight calculation by introducing two correction parameters α and β used in the final scoring calculation.

Re-ranking:

For all $i \in [\![1, n]\!]$, for each set of URIs $\{\text{URI}_i^k,\ k \in [\![1, l]\!]\}$, the re-ranking process is conducted according to the following steps:

For all $i \in I$,

1. $\forall k \in [\![1, l]\!]$, calculate dsr_score($\text{URI}_i^k$)
2. $\forall k \in [\![1, l]\!]$, calculate csr_score($\text{URI}_i^k$)
3. $\forall k \in [\![1, l]\!]$, calculate mutual_relation_score(URI_i^k) = α.dsr_score(URI_i^k)+β.csr_score(URI_i^k)
4. re-order $\{\text{URI}_i^k,\ k \in [\![1, l]\!]\}$, by decreasing order of mutual relation score.

In the following, we detail the MDP in the context of a toy example to illustrate how it works. The document contains two sentences, NE mentions are in bold:

> **IBM** has 12 research laboratories worldwide. **Thomas J. Watson, Jr.** became president of the company.

For the first NE mention [**IBM**], $\mathcal{A}_\mathcal{D}$ contains two candidate URIs identifying two different resources:

> [**IBM**] $\text{URI}_1^1 \equiv$ International Brotherhood of Magicians
> $\text{URI}_1^2 \equiv$ International Business Machines Corporation

For the second NE mention [**Thomas J. Watson, Jr.**], $\mathcal{A}_\mathcal{D}$ contains the following candidate URI, which is ranked first:

> [**Thomas J. Watson, Jr.**] $\text{URI}_2^1 \equiv$ Thomas Watson, Jr.

S_1^1 gathers URIs and categories contained in the International Brotherhood of Magicians Wikipedia page. S_1^2 is associated to the International Business Machines Corporation, and S_2^1 to the Thomas Watson, Jr. page. dsr_score(URI_1^1) sums up the number of occurrences of URI_1^1 in S_j^1 for all $j \in [\![1, n]\!] - \{1\}$. Hence, in the current example, dsr_score(URI_1^1) is the number of occurrences of URI_1^1 in S_2^1, namely the number of times the International Brotherhood of Magicians are cited in the Thomas Watson, Jr. page. Similarly, dsr_score(URI_1^2) is equal to the number of times the International Business Machines Corporation is cited in the Thomas Watson, Jr. page. csr_score(URI_1^1) sums up the number of common URIs and categories between S_1^1 and S_2^1, i.e. the number of URIs and categories appearing in both International Brotherhood of Magicians and Thomas Watson, Jr. pages. csr_score(URI_1^2) counts the number of URIs and categories appearing in both International Business Machines Corporation and Thomas Watson, Jr. pages.

After calculation, we have:

mutual_relation_score(URI_1^1) < mutual_relation_score(URI_1^2)

The candidate URIs for [**IBM**] are re-ranked accordingly, and International Business Machines Corporation becomes its first rank candidate.

4 Experiments and Results

SemLinker has been evaluated on the TAC-KBP 2012 EL task (Charton et al., 2013). In this task, mentions of entities found in a document collection must be linked to entities in a reference KB, or to new named entities discovered in the collection. The document collection built for KBP 2012 contains a combination of newswire articles (News),

	SemLinker									TAC-KBP2012 systems			
modules	no disambiguation			MDP only			all modules			1^{st}	2^{nd}	3^{rd}	median
Category	$B^{3+}P$	$B^{3+}R$	$B^{3+}F1$	$B^{3+}P$	$B^{3+}R$	$B^{3+}F1$	$B^{3+}P$	$B^{3+}R$	$B^{3+}F1$	$B^{3+}F1$	$B^{3+}F1$	$B^{3+}F1$	$B^{3+}F1$
Overall	0.620	0.633	**0.626**	0.675	0.681	0.678	0.694	0.695	**0.695**	0.730	0.699	0.689	0.536
PER	0.771	0.791	0.781	0.785	0.795	0.790	0.828	0.838	**0.833**	0.809	0.840	0.714	0.645
ORG	0.600	0.571	0.585	0.622	0.578	0.599	0.621	0.569	0.594	0.715	0.615	0.717	0.485
GPE	0.412	0.465	**0.437**	0.570	0.628	0.598	0.574	0.626	**0.599**	0.627	0.579	0.614	0.428
News	0.663	0.691	0.677	0.728	0.748	0.738	0.750	0.767	0.758	0.782	0.759	0.710	0.574
Web	0.536	0.520	0.528	0.572	0.550	0.561	0.585	0.556	0.570	0.630	0.580	0.508	0.491

Table 1: SemLinker results on the TAC-KBP 2012 test corpus with/out disambiguation modules, and three best results and median from TAC-KBP 2012 systems.

posts to blogs and newsgroups (Web). Given a query that consists of a document with a specified name mention of an entity, the task is to determine the correct node in the reference KB for the entity, adding a new node for the entity if it is not already in the reference KB. Entities can be of type person (PER), organization (ORG), or geopolitical entity (GPE). The reference knowledge base is derived from an October 2008 dump of English Wikipedia, which includes 818,741 nodes. Table 2 provides a breakdown of the queries per categories of entities, and per type of documents.

Category	All	PER	ORG	GPE	News	Web
# queries	2226	918	706	602	1471	755

Table 2: Breakdown of the TAC-KBP 2012 test corpus queries according to entity types, and document categories.

A complete description of these linguistic resources can be found in (Ellis et al., 2011). For the sake of reproducibility, we applied the KBP scoring metric ($B^3 + F$) described in (TAC-KBP, 2012), and we used the KBP scorer[1].

The evaluated system makes use of the Wikimeta annotation engine. The maximum number of candidate URIs is $l = 15$. The MDP correction parameters α and β described in Section 3.4 have been experimentally set to $\alpha = 10$, $\beta = 2$. Table 1 presents the results obtained by the system in three configurations. In the first column, the system is evaluated without the disambiguation module. In the second column, we applied the MDP without correction processes. The system with the complete disambiguation module obtained the results provided in the third column. The three best results and the median from TAC-KBP 2012 systems are shown in the remaining columns for the sake of comparison.

[1] http://www.nist.gov/tac/2013/KBP/EntityLinking/tools.html

We observe that the complete algorithm (co-references, named entity labels and MDP) provides the best results on PER NE links. On GPE and ORG entities, the simple application of MDP without prior corrections obtains the best results. A slight loss of accuracy is observed on ORG NEs when the MDP is applied with corrections. For those three categories of entities, we show that the complete system improves the performance of a simple algorithm using distance measures. Results on categories News and Web show that the best performance on the whole KBP corpus (without distinction of NE categories) is obtained with the complete algorithm.

5 Conclusion

The presented system provides a robust semantic disambiguation method, based on mutual relation of entities inside a document, using a standard annotation engine. It uses co-reference, NE normalization methods, and Wikipedia internal links as mutual disambiguation resource to improve the annotations. We show that our proposition improves the performance of a standard annotation engine applied to the TAC-KBP evaluation framework. SemLinker is fully implemented, and publicly released as an open source toolkit (`http://code.google.com/p/semlinker`). It has been deployed in the TAC-KBP 2013 evaluation campaign. Our future work will integrate other annotation engines in the system architecture in a collaborative approach.

Acknowledgments

This research was supported as part of Dr Eric Charton's Mitacs Elevate Grant sponsored by 3CE. Participation of Dr Marie-Jean Meurs was supported by the Genozymes Project funded by Genome Canada & Génome Québec. The Concordia Tsang Lab provided computing resources.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Razvan C. Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. ACL.

Eric Charton and Michel Gagnon. 2012. A disambiguation resource extracted from Wikipedia for semantic annotation. In *Proceedings of LREC 2012*.

Eric Charton, Marie-Jean Meurs, Ludovic Jean-Louis, and Michel Gagnon. 2013. SemLinker system for KBP2013: A disambiguation algorithm based on mutual relations of semantic annotations inside a document. In *Text Analysis Conference KBP*. U.S. National Institute of Standards and Technology (NIST).

Eric Charton, Marie-Jean Meurs, Ludovic Jean-Louis, and Michel Gagnon. 2014. Improving Entity Linking using Surface Form Refinement. In *Proceedings of LREC 2014*.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP-CoNLL*. ACL.

Joe Ellis, Xuansong Li, Kira Griffitt, Stephanie M Strassel, and Jonathan Wright. 2011. Linguistic resources for 2012 knowledge base population evaluations. In *Proceedings of TAC-KBP 2012*.

Angela Fahrni, Thierry Göckel, and Michael Strube. 2012. Hitsmonolingual and cross-lingual entity linking system at tac 2012: A joint approach. In *TAC (Text Analysis Conference) 2012 Workshop*.

Aldo Gangemi. 2013. A Comparison of Knowledge Extraction Tools for the Semantic Web. In *The 10th Extended Semantic Web Conference (ESWC) 2013*.

Xianpei Han and Jun Zhao. 2009. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceedings of the Conference on Information and Knowledge Management (CIKM)*. ACM.

Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard De Melo, and Gerhard Weikum. 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.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 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.

Heng Ji, Ralph Grishman, HT Dang, and K Griffitt. 2010. Overview of the TAC 2010 knowledge base population track. *Proceedings of TAC 2010*.

Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. DBpedia Spotlight: Shedding Light on the Web of Documents. In *The 7th International Conference on Semantic Systems (I-Semantics) 2011*, pages 1–8.

David N. Milne and Ian H. Witten. 2008. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceedings of the Conference on Information and Knowledge Management (CIKM)*. ACM.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10.

Hien T. Nguyen and Tru H. Cao. 2008. Named entity disambiguation on an ontology enriched by wikipedia. In *Research, Innovation and Vision for the Future, 2008. RIVF 2008. IEEE International Conference on*, pages 247–254. IEEE.

Borislav Popov, Atanas Kiryakov, Angel Kirilov, Dimitar Manov, Damyan Ognyanoff, and Miroslav Goranov. 2003. KIM – Semantic annotation platform. *Lecture Notes in Computer Science*, pages 834–849.

Michael Strube and Simone Paolo Ponzetto. 2006. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In *AAAI*, volume 6, pages 1419–1424.

TAC-KBP. 2012. Proposed Task Description for Knowledge-Base Population at TAC 2012. In *Proceedings of TAC-KBP 2012*. National Institute of Standards and Technology.