SemLinker system for KBP2013: A disambiguation algorithm based on mutual relations of semantic annotations inside a document.

Eric Charton
Polytechnique Montreal
eric.charton@polymtl.ca

Marie-Jean Meurs
Concordia University
marie-jean.meurs@concordia.ca

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

Michel Gagnon
Polytechnique Montreal
michel.gagnon@polymtl.ca

Abstract

In this paper, we present the *SemLinker* system used by the polymtl team in the English entity linking track of TAC-KBP 2013. To improve the disambiguation process, *SemLinker* re-uses and enriches the entity links provided by a generic annotation engine. The linking is done through a re-ranking process on the candidate links associated with a given named entity. This process relies on the mutual relations between all the named entities in the document.

1 Introduction

The current definition of the TAC KBP Entity Linking (EL) task is to link named entities (NEs) found at specific positions in a document collection to entities in a reference KB, or to new NEs discovered in the collection. According to this definition, a simple way of achieving the task could be hypothesized: an existing annotation engine compliant with Wikipedia URIs could annotate a document, and then, if it exists, one could retrieve a link at a given position. In fact, such a simplification of the EL task is not yet possible because of the poor quality of annotations provided by existing engines. As shown for instance in (Gangemi, 2013), even if some of the publicly released annotation engines obtain good results on mention detection and semantic disambiguation, none or very few of them seem indeed to be precise enough to challenge the performance of the best systems specifically trained for the KBP EL task. However, there is still an interest in improving annotation quality of such generic annotation engines. The main idea of our participation to the 2013 edition of KBP is to use an existing annotation engine, and apply a complementary algorithm to improve its accuracy. The improvement is achieved through a method for computing semantic relatedness between candidate entities to link, by using knowledge from the hypertext provided by the internal links of the Wikipedia encyclopedia.

The paper is organized as follows. The main components of our approach are introduced in Section 2, then we present the architecture of our system for the TAC-KBP 2013 Entity Linking task, together with additional software and resource components in Section 3. A detailed description of resource usage is presented in Section 4. The modules of our system are described in Section 5. Experiments and results in the context of the TAC-KBP 2013 EL task are reported in Section 6. We then discuss our results, and conclude in Section 7.

2 Approach

The two major novelties in the SemLinker system reside in the query pre-processing, and in the improvement of candidate annotations provided by a generic annotation engine compliant with Wikipedia URIs. By generic annotation engine, we mean an annotator able to provide Wikipedia or DBPedia URIs as links, and which is not specifically trained or built for the KBP task. A previous attempt to involve a generic, non-Knowledge-Based supervised annotator has been conducted by (Mendes et al., 2011) with the SpotLight annotator. In this attempt, Spotlight was used with limited modifications to fit the specifics of KBP tasks. Our architecture proposition is similar but involves complementary methods to improve the disambiguation process of annotations provided by the generic annotation engine.

The basic principle of SemLinker is to re-use and to improve candidate annotations provided by the Wikimeta (Charton and Gagnon, 2012) annotation engine. Using the whole document annotations provided by the Wikimeta engine, SemLinker re-ranks the candidate links assigned to a given NE mention, according to their mutual semantic relations with the other NE mentions annotated in the document.

In past years, numerous methods have been proposed to include the semantic relatedness between concepts to improve a disambiguation process. Milne in (Milne and Witten, 2008) used a machine learning method to identify significant terms within unstructured text, and to enrich them with links to the appropriate Wikipedia articles. The proposed algorithm balances the prior probability of a sense with its relatedness to the surrounding context. The commonness of a sense is defined by the number of times it is used as a destination in Wikipedia. A different approach proposed by (Hoffart et al., 2011) uses graphs to compute relatedness. The method builds a weighted graph of mentions and candidate entities, and computes a dense subgraph that approximates the best joint mention-entity mapping. (Yazdani and Popescu-Belis, 2013) presents a method for computing semantic relatedness between words or texts by using knowledge from Wikipedia. A network of concepts is built by filtering the encyclopedia articles. Two types of weighted links between concepts are considered: one is based on hyperlinks between the texts of the articles, and the other one is based on the lexical similarity between them.

If many approaches use Wikipedia to find and compute semantic relatedness of words, or to improve a named entity disambiguation (NED) task like (Strube and Ponzetto, 2006; Bunescu and Pasca, 2006), yet, these propositions rarely involve the semantic relations between annotations of a complete document, which can be discovered according to Wikipedia knowledge. In KBP 2012, (Zhang et al., 2012) proposed to calculate a *Semantic Relatedness of Candidate Entity* value, then they used it as a feature in a Support Vector Machine classifier utilized for disambiguation. Our model uses a similar approach to calculate a semantic relatedness value, but this value is used in the context of a graph exploration rather than in a machine learning based algorithm.

3 System architecture

In our work context, a query consists of a surface form, an anchor document, and the position of the surface form in the document. The pipeline processes a query as follows:

1. Reformulate the query if necessary.
2. Annotate each NE in the document with a ranked set of Wikipedia URIs (the candidates) using an external annotator compliant with Wikipedia URIs. NE category labels (PERS, ORG...) and Part Of Speech (POS) tags are also provided.
3. Re-rank candidates for each NE in the document using all the annotation layers.
4. Extract the best NE link in the document using the query definition and query expansion.
5. Once all the queries have been processed, cluster their associated NEs with no corresponding KB entries (NILs), and convert the Wikipedia URIs format to TAC-KBP node identifiers.

Figure 1 shows the SemLinker workflow and architecture. Four main modules, described in Section 5, are dedicated to these tasks:

1. Query Reformulation module
2. Mutual Disambiguation module
3. Link Extraction module
4. Clustering module

These modules relies on various resources introduced in Section 4.

4 System resources

The SemLinker system makes use of external software: Wikimeta[1] as annotator, Lucene-Search for Wiki[2] as spell checker, and corpus resources as NLGbAse[3], and a Wikipedia dump. These resources

[1]http://www.wikimeta.org

[2]https://www.mediawiki.org/wiki/Extension:Lucene-search, v2.1

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

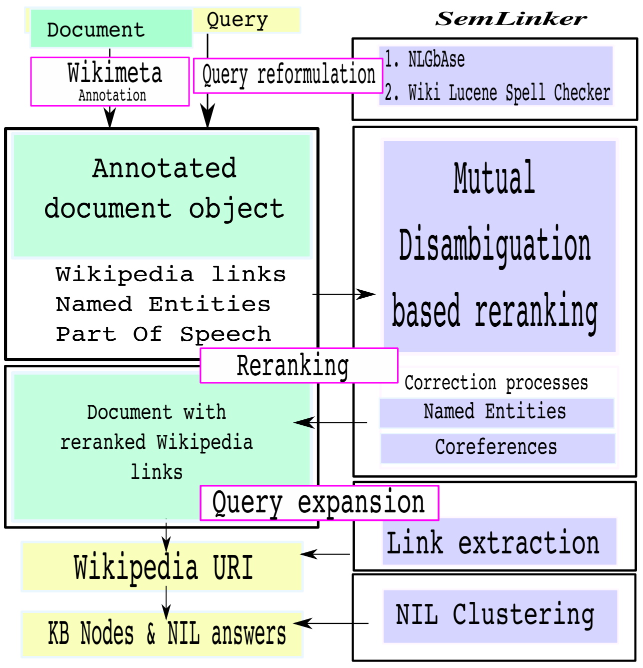


Figure 1: SemLinker workflow.

are involved in various aspects of the system like its mention correction and reformulation process, or the entity linking annotation process.

4.1 Wikipedia dump and related software

A version of Wikipedia dump was indexed with the Lucene-Search engine, and used as an internal link and category resource for the Mutual Disambiguation Algorithm. The Wikipedia dump was downloaded in July 2013. It was also utilized to train the Lucene-Search for Media-Wiki software version 2.1.3, used as a spelling correction resource for the SemLinker Query Reformulation module.

4.2 NLGbAse lexical resource

NLGbAse is a multilingual linguistic resource composed of metadata, and built from the Wikipedia encyclopedic content. The structure of Wikipedia, and the sequential process to build metadata like these in NLGbAse, have been described in (Bunescu and Pasca, 2006). The process is applied in (Charton and Torres-Moreno, 2010). For each document in Wikipedia, NLGbAse provides a set of metadata, composed of three elements: (i) a set of surface forms, (ii) all the words contained in the document, where a TF.IDF weight (Salton and Buckley, 1988) is assigned to each word, (iii) a NE tag obtained through a classification process.

The set of surface forms is obtained through the collection of every Wikipedia internal link that points to an encyclopedic document. Internal links can be *redirection links* (specific Wikipedia pages that only point to another page and express another way of spelling), *interwiki links* (links directing to the same document in another language edition) and, finally, *disambiguation pages* (pages that summarize for a unique surface form all the possibly related Wikipedia documents). For instance, the surface form set for the NE *Paris (France)*[4] contains 39 elements (eg. *Ville Lumière, Ville de Paris, Paname, Capitale de la France, Département de Paris...*). In our resource, the surface forms are collected from six linguistic editions of Wikipedia (Polish, English, German, Italian, Spanish and French). We use such cross-lingual resources because, in some cases, a surface form may appear in a single language edition of Wikipedia but should be used in other languages. For instance, the surface forms *Renault-Dacia* or *RNUR* related to a European car maker are not available in the English Wikipedia but can be collected from the Polish Wikipedia edition.

NLGbAse is involved in SemLinker as a resource for spelling correction in the Query Reformulation module. It is also used as annotation and disambiguation resource by the Wikimeta annotation engine. For this reason, NLGbAse is also utilized to build a correspondence table between Wikipedia URIs and the KB nodes (see Section 4.4).

4.3 Wikimeta annotation engine

Wikimeta is an annotation engine able to provide, for a given document, various levels of annotations. Wikimeta relies on a part-of-speech tagger, a NE recognition module, a semantic annotation module, and NLGbAse used as disambiguation interface to establish a link between annotations and the semantic web. The engine is accessible through a REST API. It is free for academic use, and its components have been described in (Charton and Gagnon, 2012).

The Wikimeta annotation engine workflow is de-

[4]http://www.nlgbase.org/perl/display.pl?query=Paris&search=FR

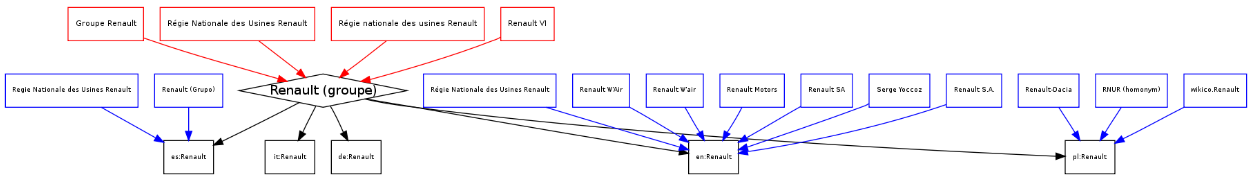


Figure 2: In this example, multiple surface forms for a company name are collected from the Polish edition of Wikipedia.

Word	POS	NE	Semantic Link
Laura	NNP	PERS.HUM	NORDF
Colby	NNP	PERS.HUM	
in	IN	UNK	
Milan	NNP	LOC.ADMI	`http://dbpedia.org/data/Milan.rdf`

Table 1: Sample annotation from the Wikimeta annotation engine. The special semantic annotation NORDF is used when no RDF link is available

scribed below.
First, NE mentions are detected using the surface forms, and a machine-learning algorithm. To label NEs as Pers, Org, Loc, Prod, Time and Date, Wikimeta uses a Conditional Random Field tagger (Béchet and Charton, 2010) to produce an initial layer of NEs. Then, Wikimeta reinforces the candidate detection process with surface forms from NLGbAse used as regular expressions to detect NEs. Finally, NEs identified in both layers are merged.

For each NE annotated in the document, the surface form is utilized to select a group of candidate Wikipedia URIs from the NLGbAse resource. A cosine similarity is calculated between the context of the NEs in the document and a bag of words with their TF.IDF for each potential candidate available in NLGbAse. The candidate links are then ranked according to their cosine scores.

One advantage of this separation between the NE labeling process and the linking annotation process is the capacity of the system to detect all NEs in the document, even if no link is available for them. For instance, *Milton*, from query EL_ENG_00309 in TAC-KBP 2012 test set, is annotated as PERS by Wikimeta with its correct complete boundaries - *Christian Milton* -, even if no link is provided by the engine. This property is very useful to manage NIL mentions as it allows, for example, to collect extended query information - the first name *Christian* in the previous sample -, and this information can be used to improve the NIL clustering process (this aspect is detailed in Section 5.4).

The Wikimeta disambiguation engine is based on the most recent version of NLGbAse, itself generated with a recent version of Wikipedia dump (January 2013). It is thus able to annotate over 3 millions of entities, far beyond the 800k present in the current KB resource. This enables our system to correctly identify many queries considered as NILs in the KBP evaluation framework. We use this complementary disambiguation capability (also used in former systems (Cucerzan, 2012; Radford et al., 2012; Graus et al., 2012)) to improve the SemLinker Clustering module effectiveness on NIL entities (see Section 5.4).

In the context of the SemLinker system, Wikimeta is utilized to annotate documents with POS labels, with NEs (including finding boundaries of NE mentions, and attributing a NE label to them), and to provide for each NE a list of ranked Wikipedia URI links. A sample of annotation is reported in Table 5. The Wikimeta annotator used for TAC-KBP 2013 was a version of the system installed in an experimental environment. It was modified to provide n candidates links (instead of 3 in the public version) for each mention detected.

4.4 Correspondence between Wikipedia URIs and KB nodes

Since Wikimeta annotations are Wikipedia URIs, we need a correspondence table between KB nodes and Wikimeta outputs. This table is generated using

NLGbAse since this resource is the Wikimeta reference knowledge base. This table is built with an algorithm that matches the exact correspondence between KB node names and NLGbAse surface forms. When a surface form of an NLGbAse entry matches exactly a KB node name, the Wikipedia URI stored in the NLGbAse entry is defined as the matching relation for the KB node. About 0.3% (less than 2.5k) of the KB nodes do not have correspondence. This is mostly due to pages deleted since 2008, that still exist in the 2008 dump used for building the KB, and that are not present in recent Wikipedia XML dumps utilized to build NLGbAse. Less than 0.1% of those missing annotations have an influence on the final process. This is controlled according to the KB nodes required in queries of the 2012 TAC-KBP test corpus that do not have matching entries in the correspondence table. We consider that this distance between our correspondence table and the KB node table does not have a notable influence on the final results.

5 SemLinker Modules

We now describe the modules involved in our pipeline to tackle the TAC-KBP EL task. The pipeline starts with a Query Reformulation module, and ends with the Clustering process of all NEs.

5.1 Query reformulation module

In EL systems, availability of an exhaustive resource of candidate surface forms is of critical interest for detecting mentions. The higher the coverage of this resource, the more candidate entities are detected. Recent TAC-KBP evaluation campaigns have been engineered to emphasize the surface form matching problem: the evaluation framework of the EL task makes increasing use of noisy and misspelled mentions that must be linked. In the 2012 and 2013 TAC-KBP evaluation corpora, we identified three main cases of mentions to annotate for which no surface form exists in Wikipedia-based resources:

- Case 1: An abbreviation that refers to a NE does not exist in any Wikipedia redirection, disambiguation or interwiki page.

  For example, a NE is denoted by an abbreviation like *JGL*, which stands for *Joseph Gordon-Levitt*[5].

- Case 2: An abbreviation that refers to a NE exists in Wikipedia, but it is redirected to another entity.

  An example is given by the *IPI*[6] surface form, that refers to *Intellectual Property Institute*. An *IPI* disambiguation page exists in Wikipedia, and allows to collect several full names for this abbreviation (see Figure 3) but does not refer to the *Intellectual Property Institute* page.

- Case 3: A mention is misspelled or provided under an uncommon or unconventional surface form, and exists in Wikipedia under a slightly different lexical description.

  This is the case of query *Bagdahd*, that should refer to the *Bagdad* page in Wikipedia[7].

These three cases cannot be handled by state-of-the-art approaches based on Wikipedia derived content only, since the surface forms collected from the encyclopedia do not match the ones expressing NEs in the document. The algorithm we propose empowers EL systems to handle such cases, and thus improves their performance.

5.1.1 Mention Correction Algorithm

We propose a Mention Correction algorithm involving two strategies to improve surface form coverage of EL systems. The first strategy consists in automatically adding supplementary surface forms generated by heuristics to an existing resource of surface forms. The second strategy involves the introduction of a lexical correction step in the surface form detection process. Let us consider a candidate mention that we want to link to a KB entry. To find a set of candidate entries in the KB according to this surface form, the proposed algorithm runs as follows:

- Step 1: The candidate mention is submitted to the Improved Surface Form Detection algorithm. If matching surface form candidates are returned, the algorithm proceeds to step 3; else to step 2.

[5]Example from query EL13_ENG_0319 of KBP 2013.
[6]Example from query EL13_ENG_1604 of KBP 2013.
[7]Example from query EL13_ENG_1872 of KBP 2013.

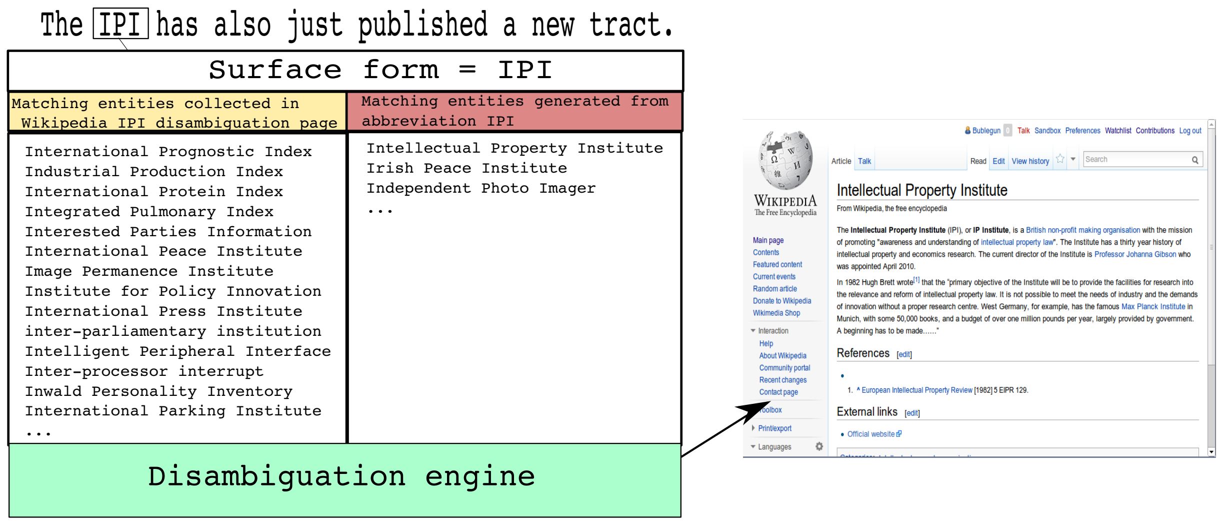

The IPI has also just published a new tract.
Surface form = IPI
Matching entities collected in Wikipedia IPI disambiguation page
International Prognostic Index
Industrial Production Index
International Protein Index
Integrated Pulmonary Index
Interested Parties Information
International Peace Institute
Image Permanence Institute
Institute for Policy Innovation
International Press Institute
inter-parliamentary institution
Intelligent Peripheral Interface
Inter-processor interrupt
Inwald Personality Inventory
International Parking Institute
...
Matching entities generated from abbreviation IPI
Intellectual Property Institute
Irish Peace Institute
Independent Photo Imager
...
Disambiguation engine
Intellectual Property Institute

Figure 3: All the abbreviated surface forms of a given entity are not necessarily present in Wikipedia. Generating some complementary matching (here for the IPI abbreviation) improves the detection capabilities of an Entity Linking system.

- Step 2: If no candidate is provided by the Improved Surface Form Detection algorithm, the candidate mention is submitted to the Surface Form Correction algorithm.
  - If this module returns suggestions of alternative surface forms, step 1 is repeated using those suggestions to collect candidates.
  - Else the algorithm returns no suggestion and exits.
- Step 3: Rewrite the query in document if needed, and proceed to annotation.

We describe below the two components of this algorithm, the Improved Surface Form Detection algorithm and the Surface Form Correction algorithm.

Improved Surface Form Detection algorithm

The original surface form resource used in this study is NLGbAse. Currently, this resource contains about 3 million English metadata, each of them describing a unique concept. The set of surface forms is obtained by collecting every Wikipedia internal link that points to an encyclopedic document. They allow matching numerous alternate spelling suggestions, including the misspelled ones (usually found in redirection pages of Wikipedia). However, this resource does not cover all the cases encountered in the TAC-KBP corpus. Hence, additional surface forms are automatically generated with various heuristics like the following ones:

- Automatic generation of abbreviations: for example, for a surface form like *Joseph Gordon-Levitt*, an heuristic generate the surface form *JGL* according to capital letters of the sequence.
- Automatic generation of alternative surface forms (adding “s” at the end of forms, reordering n-grams).

We finally obtained 4 million additional generated surface forms for a total of 14 millions, each of them related to one or more Wikipedia documents. According to step 1 of the algorithm described in Section 5.1.1, for a given word sequence in a text, this resource is used to first check if a matching surface form exists. If it exists, all related candidates are collected, and the disambiguation process is applied to rank them. If no match is provided the Surface Form Correction algorithm is invoked.

Surface Form Correction algorithm

The Surface Form Correction algorithm makes use of a database of potential spelling errors built from Wikipedia dump, and a set of rules used to validate the suggested corrections. The database engine generates a set of variations for each existing surface form. The Lucene-Wiki software is used to generate this database[8], which includes about 1 billion entries.

According to step 2 of the Query Reformulation module, the Surface Form Correction algorithm is called when the Improved Surface Form Detection algorithm did not provide any candidate. If the database engine suggests a rewriting, the resulting refined surface form is submitted to a set of selection rules intended to check if the suggestion is relevant. The rules described below are sequentially applied:

- Rule A : m common word(s).

  Let us suppose, $m = 1$, and the original surface form is *"hitlery clinton"*[9]: if the system suggests *"Hillary Rodham Clinton"*, the rule selects the suggested refined surface form because it has at least one common word with the original one.

- Rule B : Lexical distance of n letter(s).

  If $n = 1$, the original surface form is *"Michicgan"*[10], and the system suggests *"Michigan"*, the rule selects the suggested form.

- Rule C: Edit-distance between mention and suggestion.

  The rule verifies if the original and corrected mentions start with the same character and calculate their l Levenshtein distance. The suggested form is accepted if $l > t$ where t is a threshold value.

If the suggested refined form is accepted, it is submitted to the Improved Surface Form algorithm to verify if the reformulated form exits in the surface form resource.

5.1.2 Experiments on query reformulation

The algorithm was tested with the TAC-KBP 2013 English queries[11] according to the evaluation protocol of the TAC-KBP task evaluation framework. The queries consist of a surface form, an anchor document, and the position of the surface form in the document. We submitted the queries to a version of our system with Surface Form Correction disabled (refSF system in Table 2), and then enabled (QR system in Table 2). With QR system, when a refined surface form is proposed, each of its occurrences in the document is rewritten according to the new form, prior to be submitted to the semantic annotation engine. The Mention Correction algorithm improves the performance for KB link detection (KB line of Table 2), and does not reduce the performance for NILs (surface form with no matching KB link). We can conclude that the selection rules applied in the Surface Form Correction algorithm accurately reject most of the wrong corrections of surface forms. Improvement of performance obtained on the noisiest documents - DF docs of the TAC-KBP task that are web forum transcripts - also shows that the Query Reformulation module is efficient with noisy text content.

	refSF	**QR**
Category	$B^3 + F_1$	$B^3 + F_1$
Overall	0.574	**0.596**
KB (in KB)	0.494	0.535
NIL (not in KB)	0.665	0.662
NW (news doc)	0.645	0.649
WEB (web doc)	0.579	0.592
DF (forum doc)	0.454	0.508
PER (person)	0.695	0.708
ORG (organization)	0.604	0.607
GPE (geopolitical entity)	0.440	0.486

Table 2: System performance on test corpus of TAC-KBP 2013 task with reference surface form resource (refSF) and Query Reformulation (QR).

5.2 Mutual Disambiguation module

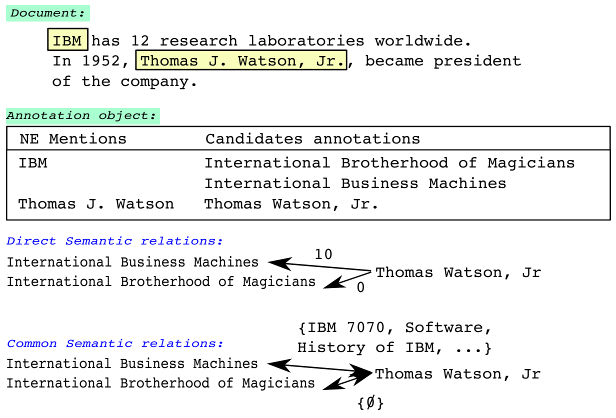
The Mutual Disambiguation module of SemLinker consists in 3 main steps:
First, it collects a set of ranked candidate Wikipedia URIs for each NE candidate in a given document using an annotator. Wikimeta is utilized for TAC-KBP 2013, but it could be any generic annotator capable of giving a list of ranked Wikipedia URIs for each NE in a document. Then, it applies simple processes of correction to improve the precision of the first rank URIs. Finally, it runs a Mutual Disam-

[8]https://www.mediawiki.org/wiki/Extension:Lucene-search

[9]Example from query EL13_ENG_0895 of KBP 2013.

[10]Exemple from query EL13_ENG_1624 of KBP 2013.

[11]`http://www.nist.gov/tac/2013/KBP/data.html`


Document:

IBM has 12 research laboratories worldwide.
In 1952, Thomas J. Watson, Jr., became president
of the company.

Annotation object:

NE Mentions	Candidates annotations
IBM	International Brotherhood of Magicians
	International Business Machines
Thomas J. Watson	Thomas Watson, Jr.

Direct Semantic relations:
International Business Machines
International Brotherhood of Magicians
10
0
Thomas Watson, Jr
Common Semantic relations:
{IBM 7070, Software, History of IBM, ...}
International Business Machines
International Brotherhood of Magicians
Thomas Watson, Jr
{∅}

Figure 4: Example of semantic relatedness captured by the Mutual Disambiguation module in SemLinker.

biguation Process (MDP) to globally improve all the annotations in the document.

The extraction of an accurate link at a specific position, according to a given TAC-KBP query, is a link extraction process occurring after the URI annotation of NEs in the whole document. Our goal here is to use all the semantic content of an annotated document to locally improve the precision of each annotation in this document. The mutual disambiguation process relies on the graph of all the relations (internal links, categories) between Wikipedia content related to the document annotations since this graph contains helpful semantic information.

A basic example of semantic relatedness that should be captured is presented in Figure 4, and 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.

We now explain the complete annotation and disambiguation process.

5.2.1 Annotation process

The document $\mathcal{D}$ is submitted to the annotation engine that provides:
- for all the words: POS tags
- for each NE found in the document: NE label, span of the NE, and links - a maximum of 15 ranked Wikipedia URIs[12].

The annotation process results in an *annotation object* called $\mathcal{A}_\mathcal{D}$. The $\mathcal{A}_\mathcal{D}$ array has one line per NE spotted in the document $\mathcal{D}$ to annotate. $\mathcal{A}_\mathcal{D}$ has $|I|$ lines, with I the index set associated with the document NEs. $\mathcal{A}_\mathcal{D}$ has 19 columns $c_{j,j\in\{1,...,19\}}$ defined as follows:

- c_1 to c_{15} store Wikipedia URIs associated with NEs, ordered by decreasing values of likelihood,
- c_{16} stores the offset of the NEs in the document,
- c_{17} stores the surface form of the NEs,
- c_{18} stores the NE labels (PERS, GPE, ...)
- c_{19} stores the POS tags.

These Wikipedia URIs are the candidate URI annotations for each NE in the document. They will be converted in KB nodes at the end of the pipeline process. When no Wikipedia URI is found by the annotation engine for the NE i at rank k and greater ($k \in \{1, ..., 15\}$), null URIs are declared in $\mathcal{A}_\mathcal{D}[i][k]$, $\mathcal{A}_\mathcal{D}[i][k+1]$,..., $\mathcal{A}_\mathcal{D}[i][15]$.

The URIs stored in colums c_1 to c_{15} of the *annotation object* will be re-ranked to improve the annotation precision. The re-ranking phase is performed by the Mutual Disambiguation Process (MDP) prepared by two Corrections Processes (CPs).

5.2.2 Correction Processes

CPs are applied before the MDP, using all the information stored in $\mathcal{A}_\mathcal{D}$. The idea is to use simple methods to locally correct some wrong annotations before applying the disambiguation process. CPs are intended to normalize the first rank URI annotations through : 1) applying the most probable NE

[12]The Wikimeta annotation engine is a specific version of the public one, installed locally, and modified to provide 15 ranked candidate Wikipedia URIs for each annotation (3 are provided in the public version)

label to NEs with similar surface forms; 2) using a co-reference algorithm based on NE labels to assign to all the NEs of a co-reference chain the same Wikipedia URI at first rank using the most probable one.

Named entity correction process

The NE correction process is applied according to the list of NEs in the *annotation object*. For two given NEs i and i' reported in $\mathcal{A}_{\mathcal{D}}$, if:

[surface form $\mathcal{A}_{\mathcal{D}}[i][17]$ is longer than $\mathcal{A}_{\mathcal{D}}[i'][17]$]
AND
[$\mathcal{A}_{\mathcal{D}}[i'][17] \subset \mathcal{A}_{\mathcal{D}}[i][17]$]

then the NE label $\mathcal{A}_{\mathcal{D}}[i][18]$ is affected to $\mathcal{A}_{\mathcal{D}}[i'][18]$.

For example, if in the document the NE *Kennedy* received the GPE label, and the NE *John F Kennedy* received the PERS label, the *Kennedy* GPE label is replaced by PERS.

Co-reference correction process

The co-reference detector is derived from (Charton et al., 2010). The co-reference detection is conducted using the information provided by the *annotation object*. Among the targeted NEs labeled as ORG, PERS, or GPE that have been detected, the ones that co-refer are identified and clustered by logical rules based on POS tags, words and NE labels. When a co-reference chain is detected, the most frequent URI associated with the longest NE surface form is attributed at first rank to all members of the chain.

5.2.3 Mutual Disambiguation Process

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 n candidate URI annotations of each mention (with $n = 15$ in the current version of SemLinker). 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.

Algorithm

Let us consider an *annotation object* $\mathcal{A}_{\mathcal{D}}$, obtained as explained in Section 5.2.1, for a document $\mathcal{D}$ of the KBP corpus.

For all $i \in I$, $k \in \{1, ..., 15\}$, we build the set S_i^k, composed of the Wikipedia URIs and categories contained in the source Wikipedia document related to $\mathcal{A}_{\mathcal{D}}[i][k]$.

Scoring:
For all $i,\ i' \in I$, $k \in \{1, ..., 15\}$, we want to calculate the weight of mutual relations between the candidate $\mathcal{A}_{\mathcal{D}}[i][k]$ and all the candidates $\mathcal{A}_{\mathcal{D}}[i'][1]$ with $i \neq i'$.

The calculation combines two scores:
- the *direct semantic relation* (dsr) score for $\mathcal{A}_{\mathcal{D}}[i][k]$ sums up the number of occurrences of $\mathcal{A}_{\mathcal{D}}[i][k]$ in $S_{i'}^1$ for all $i' \in I - \{i\}$.
- the *common semantic relation* (csr) score for $\mathcal{A}_{\mathcal{D}}[i][k]$ sums up the number of common URIs between S_i^k and $S_{i'}^1$ for all $i' \in I - \{i\}$.

Figure 4 shows an example of direct semantic and common semantic relations.

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. These parameters are used in the final scoring calculation. In the current version of SemLinker, they have been experimentally set to $\alpha = 10$ and $\beta = 2$.

Re-ranking: For all $i \in I$, for each set $\{\mathcal{A}_{\mathcal{D}}[i][k],\ k \in \{1, ..., 15\}\}$, the re-ranking process is conducted according to the following steps:

For all $i \in I$,

1. $\forall k \in \{1, ..., 15\}$, calculate dsr_score($\mathcal{A}_{\mathcal{D}}[i][k]$)
2. $\forall k \in \{1, ..., 15\}$, calculate csr_score($\mathcal{A}_{\mathcal{D}}[i][k]$)
3. calculate mutual_relation_score($\mathcal{A}_{\mathcal{D}}[i][k]$) = α.dsr_score($\mathcal{A}_{\mathcal{D}}[i][k]$) + β.csr_score($\mathcal{A}_{\mathcal{D}}[i][k]$)
4. re-rank $\{\mathcal{A}_{\mathcal{D}}[i][k],\ k \in \{1, ..., 15\}\}$ by decreasing order of mutual relation score.

5.3 Link Extraction module

The system is now ready to extract the URI associated with the TAC-KBP query. At this step we assume the majority of first rank Wikipedia URI annotations are accurate, but some errors remain. To improve the robustness of the link extraction process, our system makes use of a query expansion mechanism as described in Section 5.1. The original query or its expanded version is then submitted to heuristics that select the best final proposition of Wikipedia URI.

It is very similar in its final results as those presented for example in (Cucerzan, 2012) or (Tamang et al., 2012). Though, it differs as it can be viewed as a passive mechanism while, for example, the CUNY Blender system uses an active pattern matching to expand a GPE name that starts with only [City name] to [City name][State name]. Since the first annotation step provided by Wikimeta is a NE detection made by a CRF classifier, the returned annotated document provides for person or location names the exact boundaries of the NE surface forms without further process. In the link extraction process, this allows to automatically extend a query according to the longest surface form of the NE found at the query position.

For example, in query EL_ENG_00954, where the entity to link is *Kennedy* in the sentence *Rep. Patrick Kennedy (D-RI) has announced that ...*, Wikimeta CRF automatically finds the complete surface form *Patrick Kennedy* as a PERS. Therefore, the system only needs to collect the complete surface form related to the position of *Kennedy* in $\mathcal{A}_\mathcal{D}$ to obtain passively the expanded query *Patrick Kennedy*.

Two cases occurred in the link extraction process: 1) an expanded query is available; 2) no query expansion is available. To expand a query, the system evaluates if a NE surface form larger than the original query has been annotated by Wikimeta at the query position. If the query is an abbreviation, rules are applied to extract the corresponding full name, if it exists elsewhere in the document (e.g. the full name corresponding to the abbreviation expressed in parenthesis, just behind the first occurrence of the abbreviation). Finally, the resulting expanded query is used to filter the candidate list from $\mathcal{A}_\mathcal{D}$, and to select the best candidate according to the following rules:

- Perfectmatch : all the Wikipedia URIs for the same mention are identical (or NILs).
- BestKeyAtPos : majority of the Wikipedia URIs in the filtered list using NE label as complementary filter are identical (or NILs) with the one at the mention position.
- BestKey : majority of the Wikipedia URIs for the same mention of more than one word are identical (or NILs).
- KeyAtPos : if none of the previous heuristics gives an answer, select the Wikipedia URI (or NIL) at the mention position.

If there is no Wikipedia URI provided after application of all these rules, NIL value is assumed, but the expanded query is saved for later use in the NIL clustering process.

5.4 Clustering module

Before the final clustering process, entity links provided by our system are Wikipedia URIs. They do not take KB nodes into account. As the annotation engine uses a recent edition of Wikipedia, many NIL entities also receive a Wikipedia URI in our system. The clustering process for annotated entities is implicit, and requires no specific actions: all the queries with a Wikipedia URI link are assumed to belong to the same cluster.

Then remains a need of clustering for entities that have not received any Wikipedia URI. The clustering process for those remaining entities follows 2 steps:

1. make use of extended queries collected during the link extraction process to attach remaining queries to existing clusters, or create new ones when they share a similar surface form of more than one word,
2. remaining entities are clustered as singletons.

Finally, KB node affectation is done using the correspondence table between Wikipedia URIs and KB nodes which is described in Section 4.4. Remaining clusters receive a NIL reference.

6 Experiments

The system was developed using the TAC-KBP 2012 queries[13]. On this resource, used for training, our system obtained the results presented in Table 3. Compared to the TAC-KBP 2012 official results in Table 4, our system obtains near state-of-the art performance.

On the test corpus of TAC-KBP 2013, the score of our system is higher than the median of the global

[13]tac_2012_kbp_english_evaluation_entity_linking_query.xml

Category	**global median** $B^3 + F_1$	**SemLinker** $B^3 + F_1$	**no-web-no-wiki median** $B^3 + F_1$
Overall	0.588	**0.596**	0.540
KB (in KB)	0.535	0.535	0.488
NIL (not in KB)	0.611	0.662	0.662
NW (news doc)	0.664	0.649	0.573
WEB (web doc)	0.522	**0.592**	0.481
DF (forum doc)	0.469	0.508	0.448
PER (person)	0.610	**0.708**	0.550
ORG (organization)	0.542	**0.607**	0.510
GPE (geopolitical entity)	0.543	0.486	0.485

Table 5: TAC-KBP 2013 SemLinker results and median results of the global and no-web-no-wiki categories

Category	$B^3 + P$	$B^3 + R$	$B^3 + F1$
Overall	0.695	0.696	0.695
NIL	0.786	0.759	0.772
KB	0.635	0.639	0.637

Table 3: SemLinker results on the TAC-KBP 2012 test corpus used as development resource.

Rang $B^3 + F1$	1	2	3
Overall	0,730	0,699	0,689
NIL	0,789	0,781	0,765
KB	0,685	0,653	0,620

Table 4: Official results of the three best systems on the TAC-KBP 2012 test corpus.

results, and also higher than the median score of the no-web-no-wiki category. The median of global results is calculated on the best runs of the 26 teams, the median of the no-web-no-wiki category is calculated on the best runs of 9 teams. The results are presented in Table 5. We observe that our system obtains a good level of performance on various types of documents, and on PERS and ORG NE categories, but under-performs on the GPE NEs. After investigation, it appears that this low level of performance on this specific category of NEs is due to a heuristic applied on GPE in the link extraction process (presented in 5.3). This GPE heuristic, which works correctly on the development corpus, reduces the performance on the test corpus. This can be considered as a development error that should be solved by minimal correction for improving subsequently the global performance of SemLinker.

7 Conclusion

We have presented SemLinker, a general experimental platform dedicated to the exploration of the entity linking task according to the TAC-KBP evaluation framework. This platform can be easily adapted with various external annotation tools compliant with Wikipedia URI annotation format. We plan to implement complementary annotation software to compare them in this experimental context. Our system introduces two novel techniques to improve entity linking: a Query Reformulation module, and a Mutual Disambiguation algorithm based on semantic relatedness in a document. Using a generic annotation engine, an annotator able to provide Wikipedia URIs as links and not specifically trained or built for the TAC-KBP task, this proposition obtained an encouraging median $B^3 + F1$ score in the general task without any supervised training using the KB content or Wiki-text. Our system performs better than the median score compared to the scores of non-supervised systems (no Wiki-text). The source code of our system is publicly available. We plan to upgrade it in the next months to compare the performance of various annotation tools in the same context.

Reproducibility

All the experiments presented in this paper are fully reproducible on NIST TAC-KBP data using the SemLinker software[14].

[14]Open Source and publicly released at `https://code.google.com/p/semlinker/`

Acknowledgments

This research was supported as part of Dr Eric Charton **Mitacs Elevate** Grant, sponsored by 3CE. Participation of Dr Marie-Jean Meurs in this work was supported by the Genozymes Project, a project funded by Genome Canada and Génome Québec. Authors would like to thank Wikimeta Technologies Inc. which provided support and computing resources for this research.

References

Frédéric Béchet and Eric Charton. 2010. Unsupervised knowledge acquisition for extracting named entities from speech. In *ICASSP 2010*, Dallas. ICASSP.

Razvan C. Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL*, volume 6.

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

Eric Charton and J.M. Torres-Moreno. 2010. NLGbAse: a free linguistic resource for Natural Language Processing systems. In LREC, editor, *LREC 2010*, number 1, Malta. Proceedings of LREC 2010.

Eric Charton, Michel Gagnon, and Benoit Ozell. 2010. Poly-co : an unsupervised co-reference detection system. In *INLG 2010-GREC*, Dublin. ACL SIGGEN.

Silviu Cucerzan. 2012. The MSR System for Entity Linking at TAC 2012. In *Proceedings of the Text Analysis Conference 2012*.

Aldo Gangemi. 2013. A Comparison of Knowledge Extraction Tools for the Semantic Web. In *10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013*.

David Graus, Tom Kenter, Marc Bron, Edgar Meij, and Maarten De Rijke. 2012. Context-Based Entity LinkingUniversity of Amsterdam at TAC 2012. *TAC 2012*.

Johannes Hoffart, Mohamed A. Yosef, and Ilaria Bordino. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792.

Pablo N Mendes, Joachim Daiber, Max Jakob, and Christian Bizer. 2011. Evaluating DBpedia Spotlight for the TAC-KBP Entity Linking Task. In *Proceedings of the Text Analysis Conference - TAC KBP 2011*, pages 1–4.

David Milne and Ian H. Witten. 2008. Learning to link with Wikipedia. *Proceedings of the 17th ACM Conference on Information and Knowledge Management*.

Will Radford, Will Cannings, Andrew Naoum, Joel Nothman, Glen Pink, Daniel Tse, and James R Curran. 2012. (Almost) Total Recall -. In *Proceedings of the Text Analysis Conference 2012*.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*.

Michael Strube and Simone Paolo Ponzetto. 2006. WikiRelate! Computing Semantic Relatedness Using Wikipedia. Number February.

Suzanne Tamang, Zheng Chen, and Heng Ji. 2012. CUNY BLENDER TAC-KBP2012 Entity Linking System and Slot Filling Validation System. In *Proceedings of the Text Analysis Conference 2012*, number Cc.

Majid Yazdani and Andrei Popescu-Belis. 2013. Computing text semantic relatedness using the contents and links of a hypertext encyclopedia. *Artificial Intelligence*.

Tao Zhang, Kang Liu, and Jun Zhao. 2012. The NLPRIR Entity Linking System at TAC 2012. *Text Analysis Conference KBP*.