Open Access

Open Journal of Semantic Web (OJSW)
Volume 5, Issue 1, 2018

http://www.ronpub.com/ojsw
ISSN 2199-336X

FICLONE: Improving DBpedia Spotlight Using Named Entity Recognition and Collective Disambiguation

Mohamed Chabchoub[A], Michel Gagnon[A], Amal Zouaq[B]

[A] Polytechnique Montréal, 2900 Edouard Montpetit Blvd, Montréal, QC H3T 1J4, Canada,
{mohamed.chabchoub, michel.gagnon}@polymtl.ca
[B] School of Electrical Engineering and Computer Science (EECS), University of Ottawa,
800 King Edward Ave., K1N 6N5 Ottawa, Canada, azouaq@uottawa.ca

Abstract

In this paper we present FICLONE, which aims to improve the performance of DBpedia Spotlight, not only for the task of semantic annotation (SA), but also for the sub-task of named entity disambiguation (NED). To achieve this aim, first we enhance the spotting phase by combining a named entity recognition system (Stanford NER) with the results of DBpedia Spotlight. Second, we improve the disambiguation phase by using coreference resolution and exploiting a lexicon that associates a list of potential entities of Wikipedia to surface forms. Finally, to select the correct entity among the candidates found for one mention, FICLONE relies on collective disambiguation, an approach that has proved successful in many other annotators, and that takes into consideration the other mentions in the text. Our experiments show that FICLONE not only substantially improves the performance of DBpedia Spotlight for the NED sub-task but also generally outperforms other state-of-the-art systems. For the SA sub-task, FICLONE also outperforms DBpedia Spotlight against the dataset provided by the DBpedia Spotlight team.

Type of Paper and Keywords

Regular research paper: *semantic annotation, named entity disambiguation, DBpedia Spotlight, collective disambiguation*

1 Introduction

Semantic annotation is the task of identifying all relevant entities in unstructured textual resources and linking them to a knowledge base. The last decade has witnessed the development of several semantic annotator services, which are important tools for the evolution towards a Semantic Web. Many services use the cross-domain encyclopedia Wikipedia or more specifically, DBpedia, an RDF dataset extracted from Wikipedia. One of the most prominent semantic annotators is DBpedia Spotlight, an open-source tool available for many languages, including English, French, German, Spanish and Portuguese. It distinguishes itself from the other annotators, which are either limited to one or two languages, or only available as paid services. Yet DBpedia Spotlight's performance still requires improvement to make it really competitive with other services. The only other available open-source and multilingual semantic annotation service is Babelfy [13]. As we will show later in this paper, Babelfy is outperformed by DBpedia Spotlight, which motivates our choice of DBpedia Spotlight. Another challenge with DBpedia is its inability to distinguish two

related but different tasks:

Named entity disambiguation (NED): This task focuses on the annotation of named entities, which refer to individuals of certain types, such as *Montreal* and *Winston Churchill*. Named entity annotation is an extension of the simpler task of named entity recognition (NER), an important topic in natural language processing, which has been vastly studied and investigated in the literature [14]. The main difference is that traditional NER has very limited types such as *person* and *organization* that are generally not defined in an ontology. On top of these traditional named entities, current linked-data based annotators define an extended range of named entities and rely on a finer classification of each named entity (e.g. politicians, poets and non-governmental organizations).

Full semantic annotation (SA): Given a particular knowledge base, such as DBpedia, SA consists in the identification of all the possible entities in a document, which have an entry in the knowledge base. This includes named entities, but also abstract concepts, like *architecture*, or classes of entities, like *mayor*. The early semantic annotation platform KIM [11] is a good example of such an approach, which is often based on the assumption of a closed knowledge base.

DBpedia Spotlight only offers a service for full semantic annotation. In many applications, we are interested mainly in the identification of named entities, and DBpedia Spotlight will thus detect many additional concepts that will introduce some noise in the process. Another problem is that DBpedia also misses many relevant named entities that exist in the text.

More specifically, we will show that the two main shortcomings of DBpedia Spotlight, i.e. an unsatisfactory performance and inability to detect only named entities, can be solved using the following techniques:

- The combination of DBpedia Spotlight with a named entity recognizer, which results in an enriched set of detected entities.
- The use of an external lexical resource to identify potential candidates to be linked to the mentions for which no entity is proposed by DBpedia Spotlight, namely the additional mentions obtained from the named entity recognizer.
- A collective disambiguation process, where the decision made for one entity takes into account the decisions made for all the other entities mentioned in the text.
- A simple coreference method, which is used to recognize different ways of mentioning the same entity in the text. For instance, the entity corresponding to the mention *Michael Schumarer* at some place in the text could be designated by the mention *Schumarer* in another part of the text. This helps to avoid linking a potentially ambiguous mention, like *Schumarer*, to the wrong entity.

In this paper, we develop FICLONE, which is composed of two services based on DBpedia Spotlight: FICLONE NED, which focuses on the annotation and disambiguation of named entities, and can thus be compared to other NED systems, and FICLONE SA, which annotates and disambiguate all the entities (named entities and other concepts) in the text. The technologies and methods we used to improve the performance of DBpedia Spotlight are not new, but to the best of our knowledge, no research work attempts to combine them to improve the performances of a semantic annotator. As we will see in our experimentation results (Section 5), combining them to DBpedia Spotlight makes it competitive with the state-of-the-art systems for the NED task. Concerning the SA task, the situation is less clear, but we will see that for one of the two evaluation corpora, the SA performance of DBpedia is greatly improved.

The rest of this paper is organized as follows. In the next section, we briefly describe the main state-of-the-art systems for the SA and NED tasks. In Sections 3 and 4, we describe FICLONE NED and FICLONE SA, respectively. In Section 5 we perform an extensive evaluation on the two services, and the experimental results show how they compare favourably to the best annotators that are publicly available. In section 6, we give a thorough analysis and discussion on the evaluation results. Finally, we conclude this work in Section 7 and propose future work to further improve the performance of our FICLONE.

2 Related Work

Several approaches have been proposed to tackle the semantic annotation task. In this section we present the most well-known systems in this area. In general, the task can be divided into three main steps:

1. **Entity spotting:** to identify the most relevant mentions in the input text. A *mention* is any sequence of words or expression that is used to designate an entity in a reference knowledge base.

2. **Candidate generation:** to assign a list of candidates from a knowledge base to spotted mentions.

3. **Entity disambiguation:** to find the best candidate for each mention. Generally, the knowledge base that is used for disambiguation is DBpedia. Thus, for every mention of a certain entity in the text, there will be a corresponding URI for this entity in DBpedia.

When annotating a piece of text, DBpedia Spotlight [5] relies on shallow natural language processing tools to identify phrases that could be linked to some DBpedia entities. Once this step is completed, we obtain a set of mentions that are deemed to be relevant. For each mention, there are usually more than one candidate entity in DBpedia. A generative probabilistic model is used to select the most relevant one. This model is based on three probabilities that are estimated using a corpus of all hyperlinks found in Wikipedia. For each hyperlink, we have an entity e, which corresponds to the target of the link, an anchor text s and a context c, which is composed of the words that form the sentence or paragraph in which the hyperlink is found.

The estimated probabilities are: (1) the prior probability $P(e)$, which essentially corresponds to the ratio of occurrences of e in the set of hyperlinks, (2) the probability $P(s|e)$, which corresponds to the ratio of occurrences of s in all hyperlinks that point to e, and (3) the probability $P(c|e)$, which combines, for all words w in c, their probability of appearing in the context of a hyperlink that points to e. For a mention s in the text, whose context is c, the selected entity is the entity e that maximizes the combined probability $P(e)P(s|e)P(c|e)$.

One main limitation of DBpedia Spotlight is that it performs what we call *individual disambiguation*, i.e., each mention is annotated without considering the decision taken for the other mentions. Recently, *collective disambiguation* approaches have shown much promise to improve the performance of the task. In the following, we describe a set of semantic annotators that rely on collective disambiguation: Wikipedia Miner [12], Tagme [7], AIDA [9], WAT [17] and Babelfy [13].

In collective disambiguation, the disambiguation of a mention has an influence on the disambiguation of the other mentions in the same text. This method was first introduced in Wikipedia Miner [12]. This annotator detects the unambiguous mentions (i.e mentions that have only one candidate) and then uses these annotations to disambiguate the other mentions. It also uses a relatedness formula, which expresses how much two entities are semantically related. Various classifiers (SVM, naive Bayes classifiers and decision trees) are trained to balance between the prior score and the relatedness with other unambiguous mentions.

Tagme is another recent semantic annotator based on collective disambiguation proposed in [7]. The disambiguation is achieved in a manner similar to Wikipedia Miner, but instead of considering only the unambiguous mentions, all other mentions are used. Let entity e be a candidate for mention m. For every other mention in the text, a score is determined in relation with the entity e. The entity with the highest vote score is then selected. Tagme also implements a pruning phase to detect irrelevant mentions that should be ignored in the annotation process. WAT [17] is an enhanced version of Tagme, in which two main modifications are made. For the spotting phase, WAT uses the prior score to eliminate irrelevant mentions. For the disambiguation step, it represents the disambiguation task as a graph, where mentions and their candidates are described as nodes. The aim of the approach is to find the sub-graph that interconnects a maximum of mentions from the main graph.

A different approach was developed for AIDA [9], where Stanford NER, which is based on Conditional Random Fields (CRF) models, was used as a named entity recognizer. For each mention spotted in the text, a list of candidates is produced by looking for the YAGO [19] entities whose label matches the mention. To disambiguate the spotted mentions, AIDA uses a graph-based algorithm, where both textual mentions and candidate entities are nodes. The mention-candidate edges are weighted with contextual similarity combined with the prior score, while the relatedness defined in Wikipedia Miner is used to weight the candidate-candidate edges. AIDA extracts the sub-graph with the best density, using a combination of the three computed scores (relatedness, prior score and contextual score).

Finally, Babelfy [13] uses a part-of-speech tagger to identify relevant mentions that contain at least one entry in Babelnet [15]. A random walk algorithm is applied to discover the set of entities that are reachable from an entity e. This set of entities is called the "semantic signature" of the entity e. Similarly to AIDA, Babelfy uses a graph-based approach. Every node in the graph is represented by a pair <m, c>, where m is a spotted mention and c is one of its candidates. An edge is added from <m, c> to <m', c'> only when $c' \in semanticSignature(c)$ and $m \neq m'$. Each node is weighted with a score that computes the number of its incoming and outgoing edges. At the end, Babelfy keeps, for each mention, the candidate that has the highest score.

3 FICLONE NED

In this section, we describe FICLONE NED, a service that transforms DBpedia Spotlight into a genuine named entity disambiguator. Since DBpedia Spotlight is a general semantic annotator, the spotted mentions can contain common names, like *president* and *company*. For this reason, it does not perform well on datasets where only named entities are recognized. To fix this issue, we use Stanford named entity recognizer [8] as a tool to distinguish between a named entity and a common name. We also use this named entity recognizer to fix some spotting errors made by Spotlight and to spot relevant mentions that DBpedia Spotlight is not able to detect.

DBpedia Spotlight offers a parameter, called *confidence*, which allows users to balance between precision and recall. Assigning the value 0 to this parameter means that DBpedia Spotlight detects every possible mention, while the value 1 keeps only mentions for which DBpedia Spotlight is sure of the correctness of the linked entity. To determine which mentions are retained, DBpedia Spotlight computes the difference between the score of the best candidate entity e_1 and the second-best candidate entity e_2 for the same mention m. If their scores are close, DBpedia Spotlight returns the annotation $m \rightarrow e_1$ only if the value assigned to the confidence parameter is low. Put simply, the bigger the confidence value, the greater the score difference must be in order to keep the best candidate. DBpedia Spotlight also offers a service[1] that returns all the best candidates for every mention. As we will see later, we use this service in our collective disambiguation step. Figure 1 illustrates the architecture of FICLONE NED. We will now explain each step in detail in the following.

3.1 Spotting

To spot relevant mentions, we adopt the output of the named entity recognizer. Stanford NER [4] is based on Conditional Random Fields (CRF) models and is widely used in the development of NLP applications. In an evaluation of named entity recognizers on bibliographical texts [1] and microposts [6], Stanford NER was identified as one of the best systems. We thus decided to use it to extract relevant named entities. Our main idea was to boost the performance of DBpedia Spotlight on named entities, so we used the recommended confidence value, 0.5, to annotate the input text. At this level we have two sets of mentions: M_D, the mentions that are returned by both the named entity recognizer and DBpedia Spotlight (i.e $NER \cap Spotlight$) and M_A, the mentions that are returned only by the named entity recognizer. We ignore the mentions that are returned only by DBpedia Spotlight, since they could correspond to concepts that are not named entities. The step of separating these sets of mentions is identified in Figure 1 as *Mention filter*.

[1] http://model.dbpedia-spotlight.org/en/candidates

3.2 Candidate Generation

The mentions contained in the set M_A are not linked to any entity in the knowledge base. We thus need to identify candidate entities for these mentions. For some of these mentions, we obtain a list of candidates by using DBpedia Spotlight (the service that returns the best candidates for each mention) with confidence value at 0.0 (remember that in this case, we get many more mentions than the ones obtained at confidence 0.5).

This is not sufficient: Some mentions detected by Stanford NER are not recognized by DBpedia Spotlight 0.0, and the list of candidates returned by DBpedia Spotlight 0.0 does not always contain the right candidate, as shown in Figure 2. In this example, DBpedia Spotlight properly annotates the mention *Stefan Schumacher* with the entity *dbpedia.org/resource/Stefan_Schumacher*, while the list returned for the mention *Schumacher* does not contain the entity *dbpedia.org/resource/Stefan_Schumacher*. For this reason, we need another source of candidates.

To solve this problem, we exploit the dataset introduced by [2]. In this paper, the authors present different datasets, which contain textual segments that are linked to a set of candidates extracted from Wikipedia. A score of TF-IDF is also indicated when the textual segment comes from a Wikipedia *anchors text* (i.e, The segment of text that is associated to a wikilink). We follow their recommendation to use LRD&WAT[2] textual segments filtered at TF-IDF threshold of 2.6, which is deemed to be well suited for tasks that require high precision, as it is the case in our requirements.

Using this dataset and the candidates found by DBpedia Spotlight 0.0, we obtain a new set M_C of mentions for which we have candidates (see Figure 1). The remaining mentions, the ones that have been detected by the NER and for which we could not find any candidate, are annotated as **NIL**, to indicate that we did not find any corresponding entry in DBpedia.

3.3 Disambiguation

To disambiguate the entities in M_A (the ones spotted by Stanford NER), we first use a coreference resolution process dedicated to the identification of persons, and

[2] http://data.dws.informatik.uni-mannheim.de/dbpedia/nlp2014/lrd-wat/

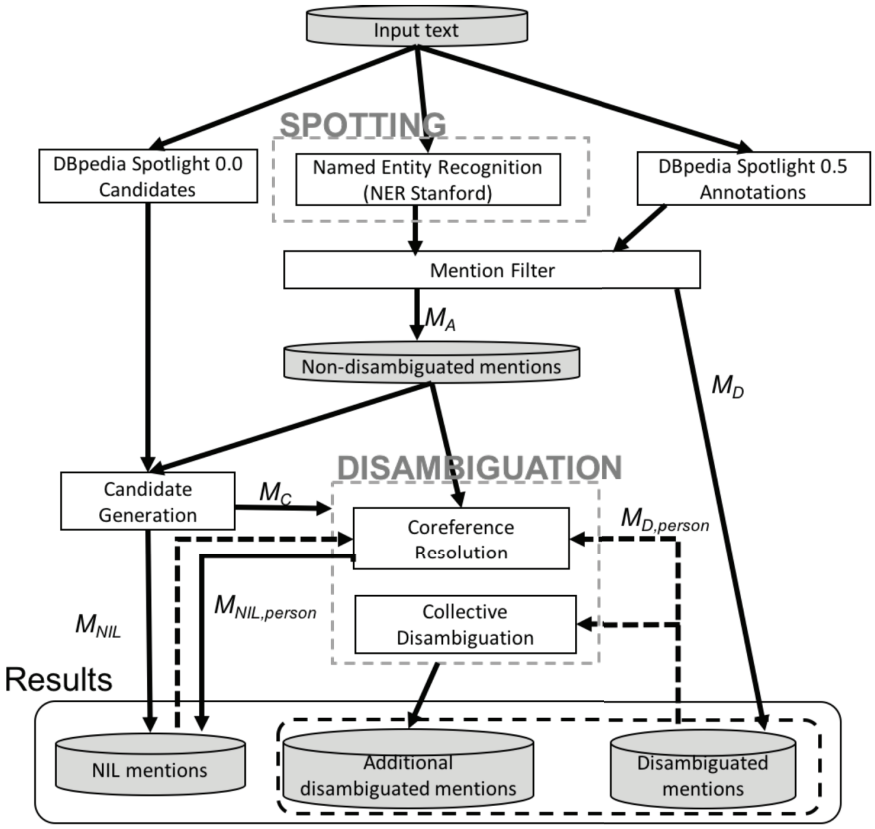


Figure 1: FICLONE NED architecture

a collective disambiguation method to disambiguate the remaining mentions.

3.3.1 Coreference Resolution

The coreference resolution method is implemented to avoid the problem of choosing the wrong candidate in cases where its name is only partially specified in the mention, whereas the full name is used in another disambiguated mention, as in the example of *Schumacher* illustrated at Figure 2. With 0.5 as confidence value, DBpedia Spotlight correctly annotates the person mentions when the full name is given in the text, like *Stefan Schumacher* $\rightarrow$ *Stefan_Schumacher* in our example. When the name is not fully specified (i.e *Schumacher*), instead of generating more candidates to this mention, we directly associate it with *Stefan_Schumacher*. To deal with this case we proceed as follows:

First, we define a subset P of DBpedia entities that are given one of the following types:

- http://dbpedia.org/ontology/Person
- http://xmlns.com/foaf/0.1/Person
- http://schema.org/Person

We exploit different namespaces because we noted in our experiments that they were complementary to identify persons. We used the set P to implement our coreference resolution process, which is decomposed into two main steps. Firstly, we identify the subset of mentions already disambiguated by DBpedia Spotlight, which correspond to persons, i.e. the set $M_{D,person} \subseteq M_D$, which contains every mention $m_i \in M_D$ that is linked to an entity $e_{person} \in P$. Secondly, we extract from M_A (remember that M_A is the set of mentions spotted by Stanford NER that are not disambiguated by DBpedia Spotlight, as shown in Figure 1) the mentions that are a substring of one of the mentions $m_k \in M_{D,person}$ and link them to the same entity as the one associated to m_k. Note that after this step, a mention may still have more than one candidate. For example, let's suppose that we have three mentions, *Schumacher* $\in M_A$, *Stefan Schumacher* $\in M_D$, linked to entity e_1 and *Elizabeth Schumacher* $\in M_D$, linked to e_2. In this case we assign both candidates e_1 and e_2 to the mention *Schumacher*.

We implemented a similar approach to deal with

Figure 2: Spotlight snapshot

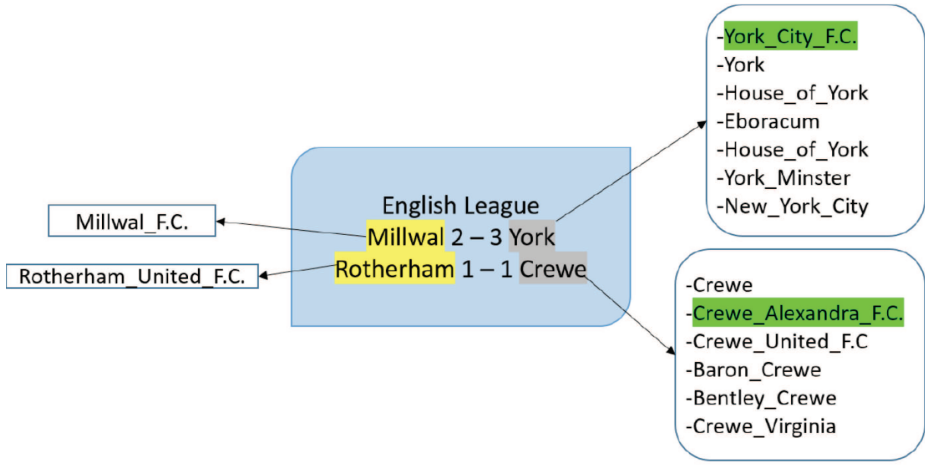


Figure 3: DBpedia Spotlight outputs

mentions annotated with NIL (set M_{NIL} in Figure 1). The NER used in FICLONE NED does not only spot named entities, it also assigns them a type (*Person*, *Organization*, *Location* and *MISC*). We use these types to identify the subset $M_{NIL,person} \in M_{NIL}$, the set of mentions that the NER identifies as a *Person*. We apply the same process described above to detect every mention $m \in M_C$ that is a substring of a mention in $M_{NIL,person}$. Since in this case there is no candidate entity, we link the mention m to NIL, and thus avoid an incorrect disambiguation with another entity that would exist in DBpedia. For example, the mention *Majed Shehadeh*$\in M_{NIL}$ has no entries in DBpedia, while the mention *Shehadeh*$\in M_C$ has several candidates, so when these two mentions appear in the same text, we annotate both of them with **NIL**.

3.3.2 Collective Disambiguation

Our main focus in this work is to enhance the performance of DBpedia Spotlight on the named entity disambiguation task, by using a collective disambiguation process. In this method, every ambiguous mention m is disambiguated by taking into account the decisions that have been made for other mentions $n \neq m$.

Consider for example the annotated snippet illustrated at Figure 3. Here we see a small table that gives the final results of two games of the English football League. As we can notice, DBpedia Spotlight correctly annotated *Millwal* with *Millwal_F.C.* and *Rotherham* with *Rotherham_United_F.C.*, but $Crewe$ and $York$ remain unlinked, since these mentions were not spotted by Spotlight. However, these last two mentions were detected by Stanford NER and we show, for each one, the list of candidates returned by the method described in the previous section. Now, the challenge is to select the best candidates (highlighted in green) by taking into account the two already disambiguated mentions (*Millwal* and *Rotherham*). Here, we should be able to consider the fact that they correspond to football clubs.

The collective disambiguation is applied to the set M_C, i.e. the set of mentions detected by Stanford NER, for which we could generate a list of candidate entities. We use in our approach two metrics: a *direct score* and a *coherence score*. The direct score corresponds to the number of times a candidate e of a mention $m \in M_C$ is linked to the entities assigned to the mentions M_D (the ones that have been disambiguated by DBpedia Spotlight). Note that after disambiguating a mention $m \in M_C$, this mention is added to the set M_D and is used to annotate other mentions from the set M_C. The coherence score is used to discriminate between entities that have the same direct score.

Our direct score, which is assigned to each candidate e_c of a mention m, is inspired from the one used in SemLinker [3] , an annotator that uses collective disambiguation. This score is based on the corresponding Wikipedia links of the entity e_c and is defined as follows:

$$DirectScore(e_c) = log(\frac{card(\{e_i | e_i \in M_D \text{ and } e_c \in links(e_i)\})}{card(M_D)} + 1) \quad (1)$$

where e_c is the candidate entity, M_D is the set of annotations that have already been disambiguated, and $links(e_i)$ is the set of links in which e_i is involved as source (outlinks) or destination (inlinks). The direct score reflects how many times a candidate e_c appears among the links of the entities that are already disambiguated. We experimented with both inlinks and outlinks. We do not report in this article all the results, but in our experiments we obtained better performances with outlinks. It seems that the occurrences of an entity e_c in the context of the entities that are already disambiguated are more relevant than the frequency of disambiguated entities in the context of e_c. We thus use only outlinks in the direct score.

Using the direct score metric is not always enough to discriminate between the entities: We observed that in some cases a set of candidates associated to the same mention have the same direct score value. As an example, for the mention "U.S." we obtained the same score for entities *United_States*, *United_States_dollar* and *United_States_Armed_Forces*. In some other cases, all entities have a direct score of 0. Another score was needed to compute the coherence of e_c with M_D. The coherence score $Coh(e_c)$ expresses how much a candidate entity is semantically related to the other entities already disambiguated. Supposing that $Sim(e_c, e_d)$ represents the relatedness of candidate e_c with an already disambiguated entity $e_d \in M_D$, the coherence score is computed by averaging over the values obtained for all entities in M_D:

$$Coh(e_c) = \frac{1}{|M_D|} \sum_{e_d \in M_D} Sim(e_c, e_d) \quad (2)$$

To calculate $Sim(e_c, e_d)$, we considered two well-known formulas: the relatedness metric introduced by [20] and the Jaccard similarity measure. The coherence score is computed only for the candidates that share the best direct score, and the rest of candidates are ignored:

$$relatedness(e_a, e_b) = \frac{log(max(|A|, |B|)) - log(A \cap B)}{log(N) - log(min(|A|, |B|))} \quad (3)$$

$$Jaccard(e_a, e_b) = \frac{(|A \cap B|)}{(|A \cup B|)} \quad (4)$$

Where e_a and e_b are two entities of interest, A and B are the sets of entities that are respectively linked to a and b, while N is the total number of entities in Wikipedia. Note that for computing the coherence score, we can consider the shared inlinks or the shared outlinks to determine whether two entities are linked.

We made some experiments to select the best combination for the coherence score, and the best results were obtained by using the Jaccard metric with outlinks. Another incentive to use outlinks is their lower number of occurrences, which makes it faster to compute the coherence metric (i.e, the entity *Canada* has 124410 inlinks and 622 outlinks).

4 FICLONE SA

In this section, we describe our approach to improve DBpedia Spotlight for the semantic annotation task, where named entities as well as other concepts are annotated. The architecture of FICLONE SA is described in Figure 4. It differs from FICLONE NED on the following aspects (indicated in red in the figure): It is not limited to the mentions spotted by Stanford NER and the mention filter is replaced by a two-step process (selection and filtering). Also, since the main purpose of the semantic annotation is to link the spotted mentions to an existing knowledge base, NIL mentions are not annotated in FICLONE SA. We detail now these differences.

Contrary to FICLONE NED (Figure 1), which uses only the output of Stanford NER, FICLONE SA also exploits the mentions detected by DBpedia Spotlight at confidence 0, thus maximizing the number of detected mentions. In our experiments we noticed that some mentions from the two sources can overlap. For example, DBpedia Spotlight may detect the mention "England football team", where Stanford NER would return only "England". In these cases, we always keep the longest mention (mention selection). We also noticed that many mentions returned by DBpedia Spotlight at confidence 0 are only adverbs, adjectives or pronouns. To fix this issue, we filter out these mentions using the Stanford POS tagger. As in FICLONE NED, the mentions disambiguated by DBpedia Spotlight 0.5 are directly added to the results. However, unlike FICLONE NED, the mentions that are not detected by Stanford NER are also added to the results (set C_D). They correspond to concepts that are not named entities.

Thus, at the end of the spotting phase, we obtain three mentions sets: M_D, the mentions that are returned by DBpedia Spotlight 0.5 and Stanford NER; M_A, the mentions returned only by Stanford NER and a new set C_D, which contains the spots returned by DBpedia Spotlight 0.5 that are not detected by Stanford NER. The last ones are not used in the collective disambiguation process, since none of these corresponds to a named entity and M_A, the set of mentions that must be disambiguated, contains only named entities. For the disambiguation step, FICLONE SA uses the same techniques as in FICLONE NED for the set M_A, namely the coreference resolution as well as the collective disambiguation process. Note that annotations in sets C_D and M_D are returned directly without further disambiguation.

5 EVALUATION

In this work, an extensive experimental evaluation on FICLONE has been performed. The results of evaluation show that FICLONE substantially improves the performances of DBpedia Spotlight. The beginning part of this section describes the methodology of the evaluation (Section 5.1). In order to compare with the FICLONE services, we first evaluate the performance of DBpedia Spotlight (Section 5.2). The evaluation of FICLONE NED and FICLONE SA are then presented in Section 5.3 and in Section 5.4 respectively. The experimental results are extensivey discussed and analized in Section 6.

5.1 Evaluation Methodology

To evaluate the two FICLONE services, we used two different kinds of datasets: NED datasets, where all the named entities are identified and SA datasets, where named entities as well as concepts are identified.

5.1.1 NED Datasets

- AIDA-CoNLL: It is a collection of 3393 news texts from *Reuters news stories* manually annotated by AIDA developers to evaluate their system [9].
- KORE 50 : This dataset is extracted from AIDA-CoNLL [10]. It contains 50 short sentences with ambiguous named entities.
- MSNBC: This dataset, introduced by [4], contains 20 news stories extracted from MSNBC. Note that only the most relevant named entities are annotated.
- N3 Reuters 128: This dataset was introduced by [18]. It contains 128 economic news articles extracted from *Reuters news stories*.

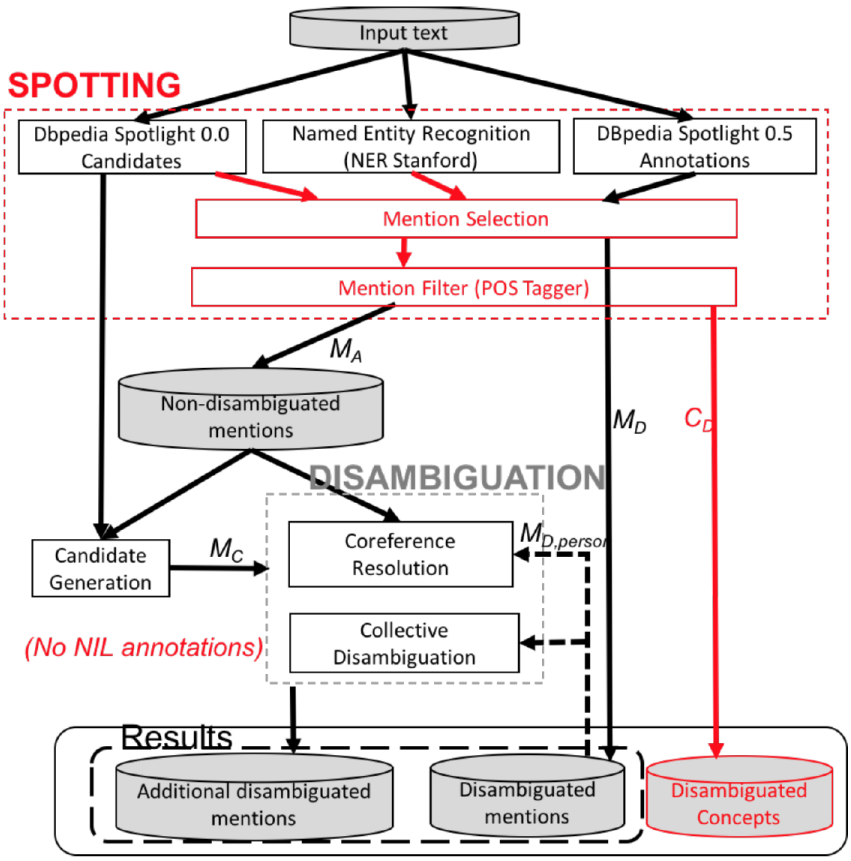


Figure 4: FICLONE SA architecture

- N3 RSS 500: This dataset was introduced by [18]. The authors extracted RSS feeds from major worldwide newspapers. They manually annotated 500 sentences randomly chosen.

5.1.2 SA Datasets

- DBpedia Spotlight dataset[3] (Spotlight DS): This dataset is composed of 58 sentences extracted from the New York Times.

- Tagme dataset[4] (Tagme DS): This dataset is composed of 180,000 short snippets of text extracted from Wikipedia 2009. In this dataset, the mentions are annotated with the Wikipedia ID page. Note that some of these IDs are not valid in the current version of Wikipedia. We thus removed from the dataset the sentences that contain a mention associated with an ID page that does not exist anymore. This reduced the dataset to 172,473 sentences. From these sentences we chose the first 10000 ones and used them for the evaluation of our system. This reduced dataset contains 47554 annotations.

Table 1 presents some descriptive statistics about these datasets. As we can notice, all the datasets contain NIL annotations for mentions that do not have any corresponding entries in the target knowledge base, except for KORE 50, Spotlight DS and Tagme DS, where NIL mentions are ignored.

Each dataset constitutes a *gold standard*, in which every mention that should be detected, together with its corresponding entity in DBpedia, is indicated. We can thus use it to compare FICLONE with other systems, by calculating the precision, the recall and the F-score. **Precision** is the ratio of mentions detected by the system, which are correct (true positives). **Recall** is the ratio of mentions in the dataset, which are correctly detected by the system. **F1-score** is the harmonic mean of precision and recall.

More formally, if N_s, N_c and N designate, respectively, the total number of mentions detected by the system, the number of mentions correctly detected

[3] yovisto.com/labs/ner-benchmarks/
[4] acube.di.unipi.it/tagme-dataset/

Table 1: Statistics on the datasets

Dataset	# Documents	# Mentions	# Disamb. mentions	# NIL mentions
AIDA-CoNLL	1393	34929	27817	7112
KORE 50	50	144	144	0
MSNBC	20	747	654	93
N3 reuters 128	128	880	631	249
N3 RSS 500	500	1000	522	478
Spotlight DS	10	329	329	0
Tagme DS	10000	47554	47554	0

by the system, and the total number of mentions in the gold standard, we can define the evaluation metrics in the following way:

$$Precision = \frac{N_c}{N_s} \tag{5}$$

$$Recall = \frac{N_c}{N} \tag{6}$$

$$F - score = \frac{2 \times P \times R}{P + R} \tag{7}$$

The three metrics have been used to evaluate the following three sub-tasks :

- Full annotation task (A2KB): a mention is counted as a true positive only when it is both correctly spotted and disambiguated.
- Entity Spotting (ES): a mention is counted as a true positive only when it is correctly spotted.
- Entity Disambiguation (ED): a mention is counted as a true positive only when it is correctly disambiguated.

Note that ED is different from A2KB, since incorrectly spotted mentions (false positives) are not considered in this evaluation.

5.2 Performances of the Baseline System – DBpedia Spotlight

Before presenting the improvement obtained by coupling DBpedia Spotlight with the methods implemented in FICLONE NED and FICLONE SA, it is important to see how DBpedia Spotlight compares to other state-of-the-art annotators. Table 2 reports the results of several systems (SA systems and NED systems) for the A2KB sub-task. Note that default configurations are used for all systems. For the Babelfy annotator, we used the NAMED_ENTITIES option and evaluated it on NED datasets.

We can notice that the two semantic annotators (i.e DBpedia Spotlight and Tagme) dominate on Spotlight DS and Tagme DS datasets, while the other ones perform better on the NED datasets (note that Babelfy's results are not impressive compared to WAT and AIDA). This confirms that semantic annotators and named entity disambiguators must be evaluated separately. It may be strange to see WAT classified as a NED system, since it is an improvement of Tagme, which is a SA system. However, as we can see in Table 2, WAT's results clearly show that it is a named entity disambiguator and is second best on NED datasets but has poor performances on SA datasets[5]. So in our evaluation, we will present the results of FICLONE NED on the NED datasets and compare it to NED systems (AIDA, WAT and Babelfy), while we test FICLONE SA on the SA datasets and compare it with SA systems (DBpedia Spotlight and Tagme).

Looking at DBpedia Spotlight, we see that its performance on NED datasets is always lower than the two best NED systems (WAT and AIDA), with the exception of N3 RSS 500, where it slightly outperforms WAT. Interestingly, for the SA task, DBpedia Spotlight obtains the best results on Tagme DS, while the situation is exactly the opposite on Spotlight DS. Looking more closely at the results of these two annotators on SA datasets, we observe that DBpedia Spotlight's weakness is its recall, which is exactly what we expect to improve with the methods implemented in FICLONE SA.

5.3 Evaluation of FICLONE NED

In this section, we present the result of several experiments. First, we show that Stanford NER greatly improves the spotting step. Second, we evaluate the performance of our collective disambiguation approach. Third, we consider the full task and compare FICLONE NED with the other state-of-the-art NED systems.

[5] In fact, by a manual inspection of the WAT's results, we noticed that only named entities are annotated.

Table 2: Performance of SA and NED systems. (For each case, we indicate precision/recall/F-score. Best and second-best values are indicated in boldface and italic, respectively.)

(a) SA systems

Dataset	Spotlight	Tagme
AIDA-CoNLL	.52/.49/.51	.19/.45/.27
KORE 50	.37/.22/.28	.30/.54/.39
MSNBC	.42/.43/.43	.11/.54/.18
N3 Reuters 128	.19/.26/.22	.05/.30/.09
N3 RSS 500	.23/.31/*.26*	.08/.36/.13
Spotlight DS	.54/.24/*.34*	.30/.60/**.40**
Tagme DS	.62/.57/**.59**	.36/.72/*.48*

(b) NED systems

Dataset	WAT	AIDA	Babelfy
AIDA-CoNLL	.62/.61/*.61*	.72/.70/**.71**	.31/.46/.37
KORE 50	.48/.43/.45	.66/.52/**.58**	.53/.55/*.54*
MSNBC	.54/.50/*.52*	.67/.60/**.63**	.26/.53/.35
N3 Reuters 128	.29/.36/*.32*	.45/.52/**.48**	.13/.28/.18
N3 RSS 500	.20/.33/.25	.43/.62/**.51**	.12/.31/.17
Spotlight DS	.27/.11/.15	.30/.11/.16	.12/.08/.10
Tagme DS	.46/.36/.40	.45/.35/.39	.29/.45/.36

Table 3: Performance of Stanford NER and DBpedia Spotlight for the entity spotting sub-task. (For each case, we indicate precision/recall/F-score.)

Dataset	Stanford NER	Spotlight
AIDA-CoNLL	.98/.97/.97	.67/.63/.65
KORE 50	.95/.87/.91	.62/.37/.46
MSNBC	.77/.79/.78	.48/.49/.49
N3 Reuters 128	.68/.81/.74	.25/.32/.28
N3 RSS 500	.58/.86/.69	.33/.42/.37

5.3.1 Impact of Using Stanford NER for Spotting

Stanford NER is a named entity recognizer: it is able to spot relevant named entities and do not disambiguate them. Thus its impact is only on the spotting step (ES sub-task). Table 3 shows the performances of DBpedia Spotlight and Stanford NER on all the selected datasets. It demonstrates the advantage of using Stanford NER in our implementation: DBpedia Spotlight's recall is much lower on all datasets. As noted earlier, DBpedia Spotlight does not distinguish between named entities and other concepts and thus detect many additional concepts, and this explains its low precision. Since Stanford NE was designed to specifically spot named entities, it is not surprising to observe that it performs much better than DBpedia Spotlight in all the datasets.

Table 4a compares the performances of FICLONE NED for spotting (which are in fact the same as Stanford NER) to the other NED systems. We can notice that the results of FICLONE NED and AIDA for the entity spotting task are the best on all the datasets. The two systems use Stanford NER, which supports our choice in using it as the main component for the spotting phase. The minor differences could be explained by the fact that FICLONE NED uses the latest version of Stanford NER while AIDA uses an oldest one.

5.3.2 Impact of Collective Disambiguation

Table 4b provides a comparison of systems' performances for the entity disambiguation sub-task (ED). These results are obtained by taking each correctly spotted mention and evaluating the correctness of the entity linked to this mention. We can see that FICLONE NED not only substantially improves the results of DBpedia Spotlight disambiguation, but also outperforms the other state-of-the-art annotators for all datasets, except KORE 50. We can also notice that FICLONE NED has the best precision for the entity disambiguation task (ED) in all the datasets except MSNBC, where DBpedia Spotlight is slightly better (0.89 vs 0.87). As expected, the good performance of FICLONE NED is mainly due to its better recall, compared to DBpedia Spotlight. Based on these results, we can conclude that FICLONE NED is more efficient in detecting the right candidate when the mention is correctly spotted.

5.3.3 Comparison to State-of-the-Art Systems for Full Task

We compared the performance of FICLONE NED with state-of-the-art annotators on all the three sub-tasks A2KB, ES and ED, defined in section 5.1. The results of A2KB are shown in Table 4c. We can observe that FICLONE NED is more competitive than DBpedia Spotlight, when compared with the other systems. The performance of FICLONE NED is the best on MSNBC and AIDA-CoNLL corpora, while AIDA is slightly better on N3 Reuters 128 and N3 Reuters 500. AIDA outperforms all the annotators on KORE 50. This dataset contains short sentences with very ambiguous mentions, thus making the task of semantic annotation very difficult.

Table 4: Comparison of FICLONE NED with other NED systems. (For each case, we indicate precision/recall/F-score. Best and second-best values are indicated in boldface and italic, respectively.)

(a) Spotting task (ES)

Dataset	FICLONE NED	AIDA	Babelfy	WAT
AIDA-CoNLL	.98/.97/**.97**	.97/.94/*.96*	.45/.65/.53	.82/.81/.82
KORE 50	.96/.87/**.91**	.94/.77/*.84*	.67/.69/.68	.87/.74/.80
MSNBC	.77/.79/*.78*	.84/.75/**.79**	.33/.66/.44	.70/.65/.67
N3 Reuters 128	.68/.81/*.74*	.74/.82/**.78**	.19/.40/.26	.37/.47/.42
N3 RSS 500	.58/.86/*.69*	.60/.84/**.70**	.21/.47/.29	.35/.53/.42

(b) Disambiguation task (ED)

Dataset	FICLONE NED	Spotlight	AIDA	Babelfy	WAT
AIDA	.77/.75/**.76**	.78/.49/.60	.74/.70/*.72*	.69/.46/.55	.75/.61/.67
KORE 50	.48/.43/.45	.40/.22/.28	.68/.52/*.59*	.69/.55/**0.61**	.52/.43/.47
MSNBC	.87/.68/**.77**	.89/.43/.58	.80/.60/*.69*	.79/.53/.63	.77/.50/.60
N3 128	.61/.53/**.57**	.53/.26/.35	.61/.52/*.56*	.51/.28/.36	.59/.36/.45
N3 500	.72/.63/**.67**	.44/.31/.36	.71/.62/*.66*	.44/.31/.36	.44/.33/.38

(c) Full task (A2KB)

Dataset	FICLONE NED	Spotlight	AIDA	Babelfy	WAT
AIDA	.75/.75/**.75**	.52/.49/.51	.72/.70/*0.71*	.31/.46/.37	.62/.61/.61
KORE 50	.46/.43/.44	.37/.22/.28	.66/.52/**0.58**	.53/.55/*0.54*	.48/.43/.45
MSNBC	.67/.68/**.68**	.42/.43/.43	.68/.60/*0.64*	.26/.53/.35	.54/.50/.52
N3 128	.43/.53/*.47*	.19/.26/.22	.45/.52/**0.48**	.13/.28/.18	.29/.36/.32
N3 500	.41/.63/*.50*	.23/.31/.26	.43/.62/**0.51**	.12/.31/.17	.20/.33/.25

5.4 Evaluation of FICLONE SA

To evaluate the performance of FICLONE SA, we compare it to Tagme and DBpedia Spotlight 0.5 on the Spotlight DS and Tagme DS datasets (A2KB, ES, ED). The results are reported in Table 5.

In Table 5a, we notice that FICLONE SA obtains the best results against the DBpedia Spotlight dataset. This is mainly due to recall, which increases substantially (from 0.24 to 0.56) for the whole annotation process. Precision decreases (from 0.54 to 0.48) mainly due to spotting, which is noisier, as shown in Table 5b (precision of 0.59 instead of 0.61). Both precision and recall are improved on the disambiguation step (see Table 5c). Now comparing to Tagme's performances on the same dataset, we also observe that FICLONE SA performs better for the full task (F-Score of 0.52 for FICLONE vs 0.40 for Tagme, according to Table 5a), mainly due to a much better precision (0.48 for FICLONE, vs 0.30 for Tagme). Based on the results on this dataset, we can conclude that the main problem of DBpedia Spotlight is its performance on recall, which is exactly the aspect that is improved in FICLONE SA.

For the Tagme dataset, our results differ. FICLONE SA's recall is improved, compared to DBpedia Spotlight, but the loss of precision is worse than on the other dataset. It seems that the decrease in precision for spotting observed in Table 5b is not compensated by the increase in precision for disambiguation (see Table 5c). This phenomenon can be explained by the fact that, contrary to Spotlight DS, not all relevant mentions are indicated in Tagme DS, thus penalizing our semantic annotator. For instance, let's consider the following text fragment:

> *... is found in caves through Kentucky and southern Indiana. It is listed as a threatened species in the United States and the IUCN lists the species as vulnerable. ...*

Mentions *United States* and *caves*, which are correctly annotated by FICLONE SA, are not indicated in Tagme DS. For this reason, we expected to observe a decrease in precision for the spotting sub-task. If we consider only the disambiguation sub-task (Table 5c), we see that the performance is improved compared to DBpedia Spotlight, as it was the case on the Spotlight DS dataset (from 0.68 to 0.74 for F-score). This indicates the good potential of the collective disambiguation process

Table 5: Comparison of FICLONE SA with other semantic annotation systems. (For each case, we indicate precision/recall/F-score. Best and second-best values are indicated in boldface and italic, respectively.)

(a) Full task (A2KB)

Dataset	FICLONE SA	Spotlight	Tagme
Spotlight DS	.48/.56/**.52**	.54/.24/.34	.30/.60/.40
Tagme DS	.41/.65/.50	.62/.57/**0.59**	.36/.72/.48

(b) Spotting (ES)

Dataset	FICLONE SA	Spotlight	Tagme
Spotlight DS	.59/.65/**.62**	0.61/.27/.37	.40/.82/.53
Tagme DS	.47/.74/.57	0.70/.63/**.66**	.43/.85/.57

(c) Disambiguation (ED)

Dataset	FICLONE SA	Spotlight	Tagme
Spotlight DS	.76/.56/.65	.65/.24/.35	.73/.60/**.66**
Tagme DS	.86/.65/.74	.85/.57/.68	.84/.72/**.78**

implemented in FICLONE SA. Tagme has the best recall and F-score for entity disambiguation on this dataset, but FICLONE SA outperforms DBpedia Spotlight and its results are closer to the results obtained by Tagme.

6 Analysis of Results

In this section, we discuss and analyze the experimental results of FICLONE against AIDA-CoNLL and MSNBC, and explain its limitations for the spotting and disambiguation steps.

6.1 Analysis of Spotting Results

To show the importance of using Stanford NER for spotting, Table 6 provides the statistics on the performances of three spotting approaches on AIDA-CoNLL and MSNBC datasets: DBpedia Spotlight 0.5, FICLONE NED (which adopts Stanford NER for spotting), and $DS \cap ST$, which is the intersection of the mentions returned by Stanford NER with the mentions returned by DBpedia Spotlight 0.5. For each, we give the total number of spotted mentions and the number of correct and incorrect spots.

Table 6 shows the problem of DBpedia Spotlight in filtering irrelevant mentions. On AIDA-CoNLL dataset, DBpedia Spotlight generates 9007 wrong mentions, out of 31758, which represents 28 per cent of the total, while using Stanford NER to filter the output of Spotlight decreases the number of wrong mentions from 9007 to 131, at the cost of losing 313 right mentions. The same case can be observed on MSNBC, where the number of irrelevant mentions decreases from 385 to 64 mentions. But using the intersection between Stanford and DBpedia Spotlight only increases the precision of DBpedia Spotlight, while using only the output of Stanford NER (the solution used in FICLONE NED) greatly increases the recall: from 22751 to 34062 correct spots on AIDA-CoNLL and from 353 to 604 on MSNBC.

6.1.1 Stanford NER Errors

As shown in Section 5.3.1, Stanford NER obtains a very high F-Score for AIDA-CoNLL dataset, while it generates a greater ratio of errors for MSNBC. Since MSNBC does not annotate all the occurrences of relevant mentions as well as the modifiers like *American, German* that are marked on the AIDA-CoNLL dataset (which causes 56 wrong spots), we focus only on cases where a mention from Stanford NER overlaps a mention in the gold standard. We found two kinds of errors:

- Mentions that should be separated (example: *Highmark Blue Cross Blue Shield of Western Pennsylvania* that should be separated into *Highmark Blue Cross Blue Shield* and *Western Pennsylvania*). We noted this kind of error 19 times on MSNBC.
- Mentions that should be enlarged (example: *University of Alabama* that should be *University of Alabama at Birmingham*). These errors occur 54 times on MSNBC.

These errors could be fixed, in our future work, by some heuristic-based technique that recognizes the

Table 6: Impact on spotting. (for each case we indicate: the number of spotted mentions / the number of correct mentions / the number of wrong mentions)

Spotting Approach	AIDA-CoNLL (34929 mentions)	MSNBC (747 mentions)
Spotlight 0.5	31758 / 22751 / 9007	738 / 353 / 385
$DS \cap ST$	22569 / 22438 / 131	368 / 304 / 64
FICLONE NED	34672 / 34062 / 610	761 / 604 / 157

Table 7: Entity disambiguation analysis. (for each case we indicate: the number of spotted mentions / the number of correct mentions / the number of wrong mentions)

disambiguation Approach	AIDA-CoNLL (27817 disamb.)	MSNBC (654 disamb.)
Spotlight 0.5	20850 / 17484 / 3366	301 / 256 / 45
Coref. resol.	1467 / 1443 / 24	110 / 108 / 2
Coherence	4790 / 3687 / 1103	114 / 85 / 29
Total	27107 / 22614 / 4493	525 / 449 / 76

composition of complex nominal phrases, such as the one proposed by [16].

6.2 Analysis of Disambiguation Results

To analyze the errors made by FICLONE in linking ambiguous mentions, once again we focus on AIDA-CoNLL and MSNBC datasets. In both FICLONE services, there are three methods of annotations: DBpedia Spotlight 0.5, the coreference resolution and the collective disambiguation process. The number of disambiguations achieved by each one of these methods are presented in Table 7, together with the number of cases where the disambiguation resulted in the selection of the correct/wrong entity. Note that we disregarded the NIL annotations from the gold standard as well as from the output of FICLONE.

The coreference resolution approach that we presented in Section 3.3.1 produced 1467 and 110 annotations with AIDA-CoNLL and MSNBC datasets, respectively, which represents 5 per cent and 21 per cent of the annotations. The coherence generates 4790 (18 per cent) and 114 (22 per cent) annotations, respectively. Together, the two methods helped to disambiguate 23 per cent of mentions in AIDA-CoNLL dataset, and 43 per cent in MSNBC, which is not negligible.

On AIDA-CoNLL, FICLONE made 4493 errors out of 27107, but we notice that 3366 of these errors come from DBpedia Spotlight 0.5. Only 1127 of the errors are due to the coreference resolution and coherence measure (this represents 25 per cent of the total). Against MSNBC, we noticed that 45 errors are made by DBpedia Spotlight, while our algorithms produced 31 errors (41 per cent of the total). This makes us to conclude that to improve the performance of FICLONE, the priority should be to find a way of correcting the annotations returned by DBpedia Spotlight.

6.2.1 FICLONE's Limitations

We noticed three kinds of problems with FICLONE, which made it less competitive in some datasets.

First, it relies completely on DBpedia Spotlight to annotate the texts. Remember that the annotations returned by DBpedia Spotlight are used directly and participate to the collective disambiguation process. Thus, wrong annotations made by DBpedia Spotlight will mislead this process. For example in Figure 5, we highlight in red, yellow and green the outputs of DBpedia Spotlight, FICLONE (the ones found by the collective disambiguation process) and the gold standard, respectively. DBpedia Spotlight wrongly annotates *Victoria* with *Victoria_(Australia)* and *Brooklyn* with *Brooklyn* (the borough of New York City) In this case, it is currently impossible for FICLONE to correctly link *David* to *David_Beckam*.

Second, FICLONE tries to annotate each mention for which it is able to extract candidates. In our example in Figure 5, we can observe that it associates *Cruz* to *Wilson_Cruz* and *Romeo* with *Romeo*, where in these cases, it should not link them to any entity. To avoid this problem, we should set a threshold for the coherence measure to filter out the low-score candidates.

Finally, in our generation of the candidate list, many possible lexicalizations are missed. For example, the mention *Tom Moody* could also be found in an abbreviated form like *T. Moody*, which are not part of the list of candidates. We would need to implement some

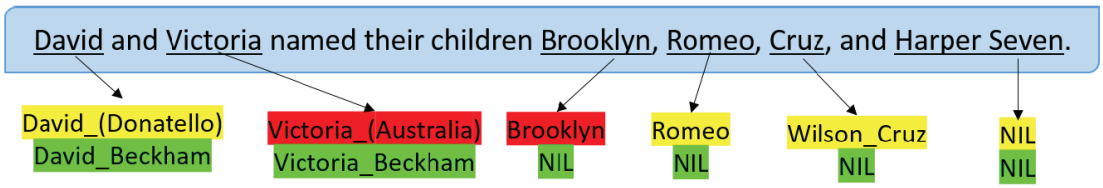


Figure 5: FICLONE errors

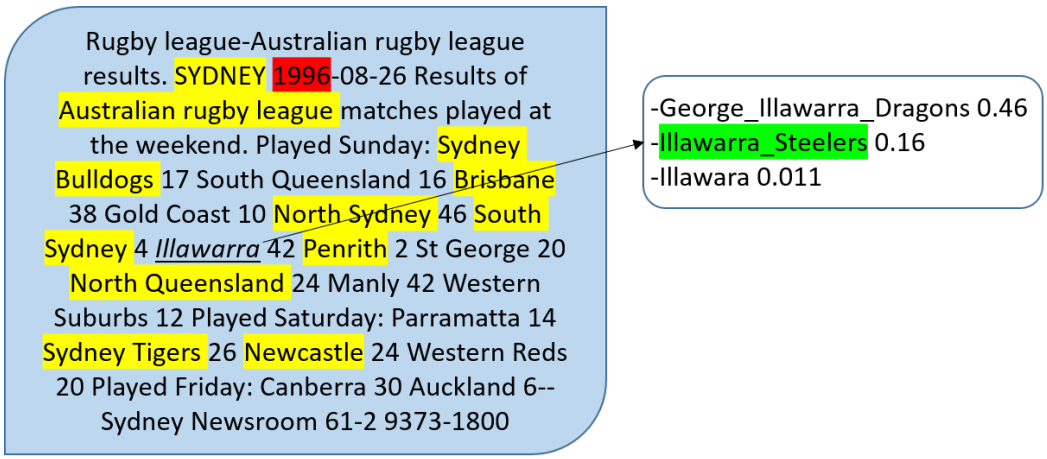


Figure 6: The difficulty of entity linking

rules to take into consideration these variations.

6.2.2 Difficulty of Disambiguation Task

The semantic annotators are usually trained on some datasets, which are not the same for all these systems. For example, in AIDA-CoNLL dataset, nationalities are annotated, (*French* is annotated with *http://dbpedia.org/resource/France*), while these mentions are ignored in MSNBC. In MSNBC, the mention *President Barack Obama* is annotated as a single mention, while in AIDA-CoNLL only *Barack Obama* is annotated. These differences clearly show that there are not any guidelines or best practices for semantic annotation datasets, and this does not facilitate the development of an annotator. How can we confidently evaluate our annotator if the available datasets do not agree on what should be annotated?

Another important challenge is illustrated in Figure 6. This example was extracted from the AIDA-CoNLL dataset. The mentions that have been already disambiguated by the semantic annotator are marked in yellow. We show the list of candidates with their score for the mention *Illawarra*. In its candidate list, we see that *St._George_Illawarra_Dragons* has the highest score (0.46), while the correct entity is *Illarawa_Steelers* (0.16). Here, we clearly see the problem of collective disambiguation: all mentions already disambiguated are related to the Australian rugby league. The two best candidates are Australian rugby teams. We cannot really expect to receive much help from collective disambiguation in this case. Other kinds of inference must be used to disambiguate this mention. Here, for example, according to the information we can obtain about the entity *St._George_Illawarra_Dragons*, this club was founded in 1998, while the results reported in the input text date from 1996. This would help determine that this can not be the correct entity.

7 Conclusion

In this paper, we showed that using a named entity recognizer for spotting of entities, and a collective approach for disambiguation of entities, substantially improves the performance of DBpedia Spotlight. For the identification of candidates that correspond to some mention in the text, we used DBpedia Spotlight *Candidates* service at confidence 0.0, combined with an external source of candidates. For the collective disambiguation process, we introduced a direct score based on the outlinks of each Wikipedia

candidate, combined with the Jaccard score used to compare each candidate to other disambiguated entities. We also demonstrated the positive impact of the coreference resolution to boost the performance of the disambiguation process. Our evaluation results show that FICLONE NED not only improves the performances of DBpedia Spotlight for the task of named entity disambiguation, but also outperforms the best semantic annotators publicly available in 4 out of 5 datasets. The experimental study also shows FICLONE SA enhances the performances of DBpedia Spotlight for the task of full semantic annotation.

For future work, we first plan to introduce linguistic methods to fix the errors made by Stanford NER. Secondly, some methods should be developed to revise the annotations made by DBpedia Spotlight before using them in our disambiguation module. A dynamic candidate generator needs to be implemented and it will take into account the context given by the text. We also plan to develop a machine learning method in order to enhance the performance of FICLONE with short texts.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] S. Atdag and V. Labatut, "A comparison of named entity recognition tools applied to biographical texts," *CoRR*, vol. abs/1308.0661, 2013. [Online]. Available: http://arxiv.org/abs/1308.0661

[2] V. Bryl, C. Bizer, and H. Paulheim, "Gathering alternative surface forms for DBpedia entities," in *NLP-DBPEDIA@ISWC*, 2015, pp. 13–24.

[3] E. Charton, M.-J. Meurs, L. Jean-Louis, and M. Gagnon, "Semlinker system for KBP2013: A disambiguation algorithm based on mutual relations of semantic annotations inside a document," in *Text Analysis Conference KBP. US National Institute of Standards and Technology (NIST)*, 2013.

[4] S. Cucerzan, "Large-scale named entity disambiguation based on Wikipedia data." in *EMNLP-CoNLL*, vol. 7, 2007, pp. 708–716.

[5] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes, "Improving efficiency and accuracy in multilingual entity extraction," in *Proceedings of the 9th International Conference on Semantic Systems*. ACM, 2013, pp. 121–124.

[6] S. Dlugolinsky, M. Ciglan, and M. Laclavik, "Evaluation of named entity recognition tools on microposts," in *Intelligent Engineering Systems (INES), 2013 IEEE 17th International Conference on*. IEEE, 2013, pp. 197–202.

[7] P. Ferragina and U. Scaiella, "Tagme: on-the-fly annotation of short text fragments (by wikipedia entities)," in *19th ACM conference on Information and knowledge management (CIKM)*, 2010, pp. 1625–1628.

[8] T. Finkel, J. R. anf Grenager and C. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," in *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005, pp. 363–370.

[9] J. Hoffart, M. A. Yosef, I. Bordino, H. Furstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum, "Robust disambiguation of named entities in text," in *8th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Stroudsburg, PA, USA, 2011, pp. 782–792.

[10] J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum, "Kore: keyphrase overlap relatedness for entity disambiguation," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 545–554.

[11] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff, "Semantic annotation, indexing, and retrieval," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 2, no. 1, 2011.

[12] D. Milne and I. H. Witten, "Learning to link with wikipedia," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 509–518.

[13] A. Moro, A. Raganato, and R. Navigli, "Entity linking meets word sense disambiguation: a unified approach," in *TACL*, 2014, pp. 231–244.

[14] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[15] R. Navigli and S. P. Ponzetto, "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network," *Artificial Intelligence*, vol. 193, pp. 217–250, 2012.

[16] E. P. A. N'Techobo, A. Zouaq, and M. Gagnon, "Semantic annotation for the analysis of political debates: A graph-based approach," in *International*

Conference on the Advances in Computational Analysis of Political Text, Dubrovnik, 2016.

[17] F. Piccinno and P. Ferragina, "From TagME to WAT: A new entity annotator," in *Proceedings of the first international workshop on Entity recognition & disambiguation*. ACM, 2014, pp. 55–62.

[18] M. Röder, R. Usbeck, S. Hellmann, D. Gerber, and A. Both, "N3 - a collection of datasets for named entity recognition and disambiguation in the NLP interchange format," *9th LREC*, 2014.

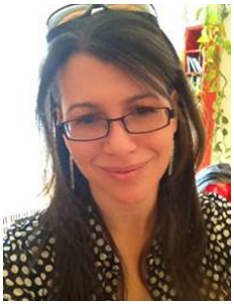
[19] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A core of semantic knowledge," in *Proceedings of the 16th International Conference on World Wide Web*, ser. WWW '07. New York, NY, USA: ACM, 2007, pp. 697–706.

[20] I. Witten and D. Milne, "An effective, low-cost measure of semantic relatedness obtained from wikipedia links," in *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, 2008, pp. 25–30.

AUTHOR BIOGRAPHIES

Mohamed Chabchoub received his bachelor degree in software engineering from the ENSI Tunisia in 2014. He completed his master degree in Computer Engineering at Polytechnique Montréal in 2016, under the supervision of Michel Gagnon and Amal Zouaq. His researches focus on automatic disambiguation of entities from plain text using the semantic web and natural language processing technologies. In 2015, his system won the first place at the Open Knowledge Extraction Challenge. He is now a software developer at Nuance Communications Inc in Montreal.

Michel Gagnon is professor at the Computer Engineering Department of Polytechnique Montreal since 2002. Previously, he worked as a team leader at Machina Sapiens inc., a company which at that time was a leader in the development of grammar checkers, and as a professor at the Univerdade Federal do Parana, in Brazil. He received his Ph.D. degree in computer science in 1993 from the Université de Montreal. He is currently working on natural language processing, with a special attention to semantics and knowledge extraction. His research activities also include the semantic web and its applications. He is co-leader of WeST Lab, whose main activities are related to the extraction of knowledge from texts.



Amal Zouaq is an Associate Professor at the University of Ottawa and an Adjunct Professor at Ecole Polytechnique de Montreal. Her research interests include natural language processing, Semantic Web, ontology engineering, knowledge extraction and technology-enhanced learning. She is the director of the WeST Lab @ uOttawa and a member of the TAMALE Lab (Text analysis and Machine Learning) at the University of Ottawa. She serves as a member of the program committee and as a reviewer in many conferences and journals in knowledge and data engineering, natural language processing, learning analytics and the Semantic Web.