EL92: Entity Linking Combining Open Source Annotators via Weighted Voting

Pablo Ruiz and **Thierry Poibeau**
Laboratoire LATTICE
CNRS, École Normale Supérieure, U. Paris 3 Sorbonne Nouvelle
1, rue Maurice Arnoux
92120 Montrouge, France
{pablo.ruiz.fabo,thierry.poibeau}@ens.fr

Abstract

Our participation at SemEval's Multilingual All-Words Sense Disambiguation and Entity Linking task is described. An English entity linking (EL) system is presented, which combines the annotations of four public open source EL services. The annotations are combined through a weighted voting scheme inspired on the ROVER method, which had not been previously tested on EL outputs. Results on the task's EL items were competitive.

1 Introduction

The paper describes our participation at SemEval 2015, Task 13 (Moro and Navigli, 2015): Multilingual all-words Sense Disambiguation (WSD) and Entity Linking (EL). Systems performing both tasks, or either one, can participate. The preferred word-sense and entity inventory is Babelnet (Navigli and Ponzetto, 2012); other inventories are allowed. Our system performs English EL to Wikipedia, combining the output of open-source, publicly available EL systems via weighted voting. The system is relevant to the task's interest in comparing the results of EL systems that apply encyclopedic knowledge only, like ours, and systems that jointly exploit encyclopedic and lexicographic resources for EL.

The paper's structure is the following: Section 2 discusses related work, and Section 3 describes the system. Sections 4 and 5 present the results and a conclusion.

2 Related Work

General surveys on EL can be found in (Cornolti et al., 2013) and (Rao et al., 2013). Work on combining NLP annotators and on evaluating EL systems is particularly relevant for our submission.

The goal of combining different NLP systems is obtaining combined results that are better than the results of each individual system. Fiscus (1997) created the ROVER method, with weighted voting to improve speech recognition outputs. A ROVER was found to improve parsing results by De la Clergerie et al. (2008). Rizzo et al. (2014) improved Named Entity Recognition results, combining systems via different machine learning algorithms. Our approach is inspired on the ROVER method, which had not been previously attempted for EL to our knowledge. Systems that combine entity linkers exist (NERD, Rizzo and Troncy, 2012). However, a difference in our system is that the set of linkers we combine is public and open-source. A second difference is the set of methods we employed to combine annotations.

EL evaluation work (Cornolti et al., 2013), (Usbeck et al., 2015) has highlighted to what an extent EL systems' performance can differ depending on characteristics of the corpus. This motivates testing whether different EL systems, properly combined, can complement each other.

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 355–359,
Denver, Colorado, June 4-5, 2015.

3 System Description

The system performs English EL to Wikipedia, combining the outputs of the following EL systems: Tagme 2[1] (Ferragina and Scaiella, 2010), DBpedia Spotlight[2] (Mendes et al. 2011), Wikipedia Miner[3] (Milne and Witten, 2008) and Babelfy[4] (Moro et al. 2014). Babelfy outputs were only considered if they started with a *WIKI* prefix or their first character was uppercase.[5] Details about each of our workflow's steps follow.

3.1 Individual Systems' Thresholds

First of all, a client requests the annotations for a text from each linker's web-service, using the services' default settings except for the confidence threshold, which is configured in our system. Annotations whose confidence is below a threshold are eliminated.

All of the linkers used, except Babelfy, output confidence scores for their annotations. Cornolti et al., (2013) reported optimal confidence-score thresholds for all our linkers (except Babelfy). Using Cornolti's BAT Framework, we verified that the thresholds are still valid.[6] We adopted the weak-annotation match thresholds for the IITB dataset, since we consider the IITB corpus close to the task's data, in text-length and topical variety. Our thresholds were 0.102 for Tagme, 0.023 for Spotlight, and 0.219 for Wikipedia Miner. Since Babelfy does not output confidence scores, all of its annotations were accepted to the next step in the workflow.

3.2 Ranking the Systems to Combine

Our method for combining annotators' outputs requires the annotators to be previously ranked for precision on an annotated reference set. It is not viable to annotate a reference set for each new corpus. To help overcome this issue, we adopt the following heuristic: We have ranked the annotators on a series of very different reference corpora. To perform EL on a new corpus, our heuristic considers the following criteria: First, the types of EL annotations needed by the user. Second, how similar the new corpus is (along dimensions described below) to the reference corpora on which we have pre-ranked the annotators. To apply the workflow to a new corpus, the heuristic chooses the annotator-ranking obtained with the reference corpus that is most similar to that new corpus, while still respecting the annotation-types needed by the user.

The reference corpora on which we pre-ranked the annotators are AIDA/CoNLL Test B (Hoffart et al., 2011), and IITB (Kulkarni et al., 2009). These corpora are very different to each other, in terms of character length, topical variety, and regarding whether they annotate common-noun mentions or not. Moreover, some EL systems obtain opposite results when evaluated on AIDA/CoNLL B vs. IITB, as tests by Cornolti et al. (2013) and on the GERBIL platform[7] have shown.

The heuristic's first criterion is the types of annotations needed: If the user needs annotations for common-noun mentions, the IITB ranking is used, since IITB is the only one in our reference-datasets that was annotated for such mentions. If the user does not need common noun annotations, our heuristic compares the user's corpus with our two reference corpora in terms of character length and of a measure of lexical cohesion. Both factors have been argued to influence linkers' uneven results across corpora (Cornolti et al., 2013).

We accepted common-noun annotations for the task, as they were relevant for the task's domains (e.g. disease names for the biomedical texts). Accordingly, the heuristic ranked annotators as per their IITB results: 1st Wikipedia Miner (0.568 precision), 2nd Babelfy (0.493), 3rd Spotlight (0.462), 4th Tagme (0.452).[8]

3.3 Weighting and Selecting Annotations

Using the linker ranking from the previous step, the annotations are voted, and selected for final output or rejected based on the vote. We used two

[1] http://tagme.di.unipi.it/tagme_help.html
[2] https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki
[3] http://wikipedia-miner.cms.waikato.ac.nz/
[4] http://babelfy.org/download.jsp
[5] Babelfy was a late addition to our pipeline; the reader will note that we made some ad-hoc decisions to benefit from its outputs while complying with previously defined features in our workflow.
[6] https://github.com/marcocor/bat-framework
[7] See http://gerbil.aksw.org/gerbil/overview at the site for the GERBIL platform (Usbeck et al., 2015):
[8] The precision is from tests in Cornolti et al., 2013, using weak-annotation-match. Babefly was not tested. In order to be able to rank it, instead of its precision we assigned it the average of all other annotators' precisions.

voting schemes. The first one relies on each annotation's confidence score, weighted by the annotator's rank and precision on the ranking datasets from 3.2. The rationale is that a high-confidence annotation for a low-ranked annotator can be better than a low-confidence annotation for a higher-ranked annotator. The definition is in Figure 1: For each annotation *(m, e)* in the results, *m* is its mention,[9] *e* is the entity paired with *m*, and Ω_m is the set of annotations in the results whose mentions overlap[10] with *m*. If the size of Ω_m is 1, the scaled confidence[11] o_{scf} of Ω_m's unique annotation *o* must reach threshold t_{uniq} in order for *o* to be accepted. Threshold t_{uniq} is the average of the scaled confidence scores for all annotations in the corpus. If Ω_m has more than one annotation, the voting is thus: For each annotation *o* in Ω_m, *o*'s vote is a product determined by several factors: o_{scf} is *o*'s scaled confidence.[12] *N* is the total number of annotators we combine (i.e. 4). Operand ro_{ant} is the rank of annotator o_{ant}, which produced annotation *o*. Po_{ant} is that annotator's precision on the ranking reference corpus (3.2 above). For ro_{ant}, 0 is the best rank and $N - 1$ the worst. Parameter α influences the distance between the annotations' votes based on their annotators' rank, and was set at 0. The annotation with the highest vote in Ω_m is accepted; the rest are rejected.

for each set Ω_m of overlapping annotations:
  if $|\Omega_m| = 1$
    for $o \in \Omega_m$: if $o_{scf} \geq t_{uniq}$ accept *o*
                          else reject *o*
  else
    select $\max_{o \in \Omega_m} \left[(o_{scf} \cdot (N - (ro_{ant} - \alpha)) \cdot Po_{ant} \right]$

Figure 1: Annotation voting scheme used in Run 1.

[9] The string of characters in the text that the annotation is based on (the term *mention* is often used in EL for this notion).

[10] Assume two mentions (p1, e1) and (p2, e2), where p1 and p2 are the mentions' first character indices, and e1 and e2 are the mentions' last character indices. The mentions overlap iff ((p1 = p2) ∧ (e1 = e2)) ∨ ((p1 = p2) ∧ (e1 < e2)) ∨ ((p1 = p2) ∧ (e2 < e1)) ∨ ((e1 = e2) ∧ (p1 < p2)) ∨ ((e1 = e2) ∧ (p2 < p1)) ∨ ((p1 < p2) ∧ (p2 < e1)) ∨ ((p2 < p1) ∧ (p1 < e2)).

[11] Since the range of confidence-scores output by each annotator was different, we minmax-scaled all original (*orig*) confidence scores to a 0-1 range: *scaled_confidence* = *(*orig*_confidence – corpus_min_orig_confidence)* / *(corpus_max_orig_confidence – corpus_min_orig_confidence)*

[12] As Babelfy does not provide confidence scores, its annotations were assigned the average over the whole result-set of the scaled confidence-scores output by the other annotators.

The second voting scheme is similar to the ROVER method in (De la Clergerie et al., 2008). The method assesses annotations based on how many linkers have produced them, using the linkers' rank, and their precision on the ranking-sets, as weights. If enough lower-ranked annotators have linked to an entity, this entity can win over an entity proposed by a higher-ranked annotator.

The voting is defined in Figure 2. For each annotation (mention *m*, entity *e*), Ω_m is the set of annotations whose mentions overlap[10] with *m*. Based on the different entities in Ω_m's annotations, Ω_m is divided into disjoint subsets, each of which contains annotations linking to a different entity. Each of these subsets *L* is voted by *vote(L)*. In *vote(L)*, for each annotation *o* in *L*, terms *N*, ro_{ant}, α, Po_{ant} have the same meaning as the terms bearing the same names in Figure 1, and are described above.

for each set Ω_m of overlapping annotations:
 for $L \in \Omega_m$:
  $vote(L) = \frac{\sum_{o \in L} \left(N - (ro_{ant} - \alpha)\right) \cdot Po_{ant}}{N}$
 if $\max_{L \in \Omega_m} (vote(L)) > P_{max}$: select $\operatorname{argmax}_{L \in \Omega_m}(vote(L))$

Figure 2: Entity voting scheme used in Runs 2 and 3.

The entity for the subset *L* which obtains the highest vote among Ω_m's subsets is selected if its vote is higher than P_{max}, i.e. the maximum precision in the ranking dataset (0.568, see Section 3.2). After selecting the winning entity, we still need to select a mention for it. The mention is selected at random among the mentions of the annotations in the winning subset *L*. This implementation of mention selection is meant as a baseline that can be refined in the future. Two initial factors to consider in mention selection would be mention length and the annotators having chosen each mention.

3.4 Entity Classification

After the vote, entities in the selected annotations are classified before final output. The classification is rule-based. It exploits the category or type labels output by the EL services we combined—except Babelfy, which does not output such information.

The classification-rules are based on type labels in the NERD ontology (Rizzo and Troncy, 2012)[13]

[13] http://nerd.eurecom.fr/ontology/nerd-v0.5.n3

and on a subset of the DBpedia ontology classes (Mendes et al. 2011)[14] relevant for the task's domains. For types *Person*, *Location*, *Organization*, Wikipedia category labels were also exploited.

Some rules involve an exact match against the annotations' categories or types, e.g. "Assign type *Location* if the annotation has type *DBpedia:Place*". Some rules involve a partial match, e.g. "Assign type *Person* if one of the Wikipedia category labels for the entity contains *births*".

For Babelfy outputs, Wikipedia category labels and DBpedia types were obtained through Wikipedia Miner's[3] and DBpedia's[15] APIs.

4 Results and Discussion

Since the task was open to systems doing either WSD or EL, or both, the corpus targeted both WSD and EL. Participant systems were evaluated on a different set of items depending on their nature (EL only, WSD only, both). The corpus contained 4 generic and domain-specific documents with 1094 single-word instances, 82 multi-words and 86 named entities (NE).

Our system was conceived and evaluated as an EL system. Table 1 shows our precision, recall and F1 for all three runs. Column *TopF1* is the maximum F1 attained by a participant on the EL items.

EL	P	R	F1	TopF1
Run1	**100**	**75.6**	**86.1**	88.9
Run2	98.3	66.3	79.2	
Run3	**100**	66.3	79.7	

Table 1: English EL results for all domains.

Run 1 results were competitive, ranking 3rd of 10, if we compare all participants' best runs. Runs 2 and 3 lag behind, due to lower recall. Run 1 employed the voting scheme in Figure 1. Runs 2 and 3 correspond to the scheme defined in Figure 2, with parameter α set to 0 in Run 2 and to 1 in Run 3. In spite of its results, the voting scheme from Figure 2 has advantages over the first one: It does not require confidence scores, so it accommodates linkers that don't score their annotations. Also, it does not need a separate threshold to decide on annotations produced by one annotator only. More work is needed to determine the reason for this difference in results, i.e. whether the second approach itself is not useful to combine EL annotations, or whether its worse results were related to our implementation.

One of the task's purposes was to compare systems' performance across domains. Table 2 shows our best run's results per domain. Column *N* reflects the number of EL items in the corpus for each domain. All other columns have the same meaning as in Table 1, but considering the per-domain results.

	N	P	R	F1	TopF1
Biomedical	48	100	83.3	90.9	100
Math & Computer	22	100	54.4	70.6	74.3
General	16	100	81.3	89.7	90.3

Table 2: English EL Run 1 results by domain.

Note that the small number of EL items available for each domain limits in our opinion the reliability of interpretations for these results.

Since our workflow combines several EL systems, it would be interesting to compare results for each individual system by itself vs. the results for the combined system. In later work (Ruiz and Poibeau, 2015), using an improved version of the system described here, and larger EL golden-sets, we performed such comparisons, finding significant improvements in the combined system vs. the individual ones.

5 Conclusion

The entity linking (EL) system presented was ranked 3rd (out of 10) on the task's EL items. The system combines the outputs of four public open source EL services. Two weighted voting methods were described to combine the outputs. The first method relies on annotations' confidence scores; the second one is a weighted majority vote. The first method obtained better results, but the second one has the advantage of being easily applicable to non-scored annotations. More work is needed to assess the reasons for the methods' differential performance. Future work also includes adding other public open source systems to the workflow.

Acknowledgments

Pablo Ruiz was supported by a PhD scholarship from Région Île-de-France.

[14] http://mappings.dbpedia.org/server/ontology/classes/
[15] http://dbpedia.org/sparql

References

Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. (2013). A framework for benchmarking entity-annotation systems. In *Proc. of WWW*, 249–260.

Éric Vilemonte De La Clergerie, Olivier Hamon, Djamel Mostefa, Christelle Ayache, Patrick Paroubek, and Anne Vilnat. (2008). Passage: from French parser evaluation to large sized treebank. In *Proc. of LREC 2008*, 3570–3576.

Paolo Ferragina and Ugo Scaiella. (2010). Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proc. of CIKM'10*, 1625–1628.

Jonathan G Fiscus. (1997). A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding, 1997*, 347–354.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. (2011). Robust disambiguation of named entities in text. In *Proc. of EMNLP*, 782–792.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. (2009). Collective annotation of Wikipedia entities in web text. In *Proc. ACM SIGKDD*, 457–466.

Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. (2011). DBpedia spotlight: shedding light on the web of documents. In *Proc. of the 7th Int. Conf. on Semantic Systems, I-SEMANTICS'11*, 1–8.

David Milne and Ian H. Witten. (2008). An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proc. of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy,* 25–30

Andrea Moro and Roberto Navigli (2015) SemEval-2015 Task 13: Multilingual All-Words Sense Disambiguation and Entity Linking. In *Proc. of SemEval-2015.*

Andrea Moro, Alessandro Raganato, and Roberto Navigli. (2014). Entity Linking meets Word Sense Disambiguation: A Unified Approach. *Transactions of the ACL*, 2, 231–244.

Roberto Navigli and Simone Ponzetto. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193, 217–250.

Delip Rao, Paul McNamee, and Mark Dredze. (2013). Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, Multilingual Information Extraction and Summarization*, 93–115. Springer.

Giuseppe Rizzo and Raphaël Troncy. (2012). NERD: a framework for unifying named entity recognition and disambiguation extraction tools. In *Proc. of the Demonstrations at EACL'12*, 73–76.

Giuseppe Rizzo, Marieke van Erp, and Raphaël Troncy. (2014). Benchmarking the Extraction and Disambiguation of Named Entities on the Semantic Web. In *Proc. of LREC 2014*, 4593–4600.

Pablo Ruiz and Thierry Poibeau. (2015). Combining Open Source Annotators for Entity Linking through Weighted Voting. In *Proceedings of *SEM 2015. Fourth Joint Conference on Lexical and Computational Semantics*. Denver, U.S.

Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga, Ciro Baron, Andrea Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccino, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. (2015). GERBIL–General Entity Annotator Benchmarking Framework. In *Proc. of WWW*.