# CEA LIST Participation at TAC EDL English Diagnostic Task

**Romaric Besançon, Hani Daher, Hervé Le Borgne, Olivier Ferret, Anne-Laure Daquo, Adrian Popescu**
CEA, LIST, Laboratory of Vision and Content Engineering, France
{romaric.besancon,hani.daher,herve.le-borgne,olivier.ferret,anne-laure.daquo,adrian.popescu}@cea.fr

## Abstract

We participated to the Entity Discovery and Linking track of TAC 2015. Since this is our first participation, we tested our system on the monolingual English diagnostic task. The purpose of this participation was to test the possibility to use the structure of the database and more specifically the relations between the entities, to improve the entity linking, in particular in the case where there is no textual description for the entity in the knowledge base.

## 1 Introduction

The goal of the Entity Discovery and Linking task in the TAC 2015 campaign (Ji et al., 2015) is to extract named entity mentions from English, Spanish or Chinese texts and link them to entities existing in a knowledge base. For our first participation, we focused only on the linking part of the task (finding the correct entity in the knowledge base knowing the entity mention), for monolingual English text. In TAC EDL 2015, the reference knowledge base is a sample of Freebase, which introduces two new features. First, the developed system must deal with the challenge represented by the very large number of entities present in the knowledge base. Second, the entity linking systems usually exploit features associated with the entity that are either context-independent (such as string matching similarities) or context-dependent (Shen et al., 2015). The context-dependent features rely on a textual description of the entity (generally the content of its Wikipedia page). However, in Freebase, all entities do no come from Wikipedia and, therefore, are not always associated with a textual description. To tackle this problem (and generally try to improve the entity linking), we propose to add a context-dependent feature that takes into account the *relation context* of the entity in the knowledge base, i.e. the entities that are in relation with the candidate entity.

We present in the following sections a more detailed description of our system and some evaluation results on both the DBpedia datasets used in previous TAC entity linking tracks and on the Freebase datasets of TAC 2015. We also discuss some error analysis we performed on these results.

## 2 System Description

### 2.1 Overview

In our participation, we want to test if using the relations between the entities in the knowledge base could improve the results of entity linking. We use a simple approach for entity linking, performing the task independently on each query. The design of our system is quite standard (Ji et al., 2014): for each query, our system performs three steps: (1) analyze the query (entity mention and textual context) (2) generate candidate entities from the knowledge base (3) select the best entity among the candidates. These steps are presented in more details in the following sections. We did not develop specific strategies for the last step required, to cluster the NIL entities: we used a simple clustering based on the string similarity of the entity mentions of the queries.

### 2.2 Knowledge Base

In TAC EDL 2015, the knowledge base used is built from a Freebase snapshot. First, a filter is applied to exclude all the entities having one of the following types: *book.written_work, book.book, music.release, music.album, tv.tv_series.episode music.composition music.recording, film.film and fic-*

*tional_universe.fictional_character*: after filtering, around 8 million entities are left. We then imported the data ($subject \xrightarrow{Predicate} object$ facts) into a relational database. The *subject* corresponds to an entity and is inserted in a table where each record is composed of the following attributes: the unique Wikipedia page title, the Wikipedia page id, the most notable type of the entity, the name in English and a tf-idf bag-of-words vector representation of the Wikipedia page associated with the entity. The *object* can be either:

- **a literal:** the fact represent a property of the entity. It is stored in a table where each record is composed of three attributes: the subject identifier, the predicate type and the alphanumeric or numeric string attribute;
- **an entity:** the fact represents a binary relation between two entities. It is stored in a junction table where each record is composed the following attributes: the subject and object identifiers (entity identifiers in the entity table) and the predicate type;
- **a compound value type (CVT):** a CVT represents a *n-ary* relation which associates an entity with several other objects, that can be entities or literal attributes. They are stored in a CVT table, whose records are composed of the CVT identifier and its type. The relations are modeled by two tables: a junction table between the CVT and entity tables, composed of the following attributes: the CVT identifier, the predicate type and the object identifier (which is an entity identifier) and a CVT literal table, used to hold the relations where the objects are literal values.

Finally the aliases and translations of every entity are inserted in a table where every record is composed of: entity identifier, alias or translation and language.

### 2.3 Query Analysis

In the diagnostic task, each query is composed of an entity mention and the document in which this mention appears. We only considered the named entity mentions (NAM) and ignored the nominal mentions (NOM), since we did not include a co-reference resolution step for query analysis. In the query analysis step, we use the document to enrich the query, both for entity mention expansion and for context representation.

More precisely, two kinds of expansion are performed, using named entities extracted from the document text by the MITIE tool[1]:

- if the entity mention is an acronym, we search in the document named entities with matching initials and add them as variants of the entity mention;
- named entity mentions whose expression includes the target entity mention are added as variants of the entity mention.

For context representation, a tf-idf vector representation of the document is built, in the same vector space as the Wikipedia documents from the knowledge base.

### 2.4 Candidate Generation

Candidate entities are generated by comparing one of the forms of the query mention (either the direct entity mention or one of its variants found by acronym expansion or named entity similarity) and the KB entities using either (Dredze et al., 2010):

1. string equality with the normalized name of the entity in the knowledge base;
2. string equality with a variation (either an alias or a translation) of an entity in the knowledge base;
3. approximate string matching with a variation of the entity in the knowledge base. In the submitted run, we use a simple string inclusion (the entity in the KB contains the targeted entity mention), since this functionality is directly available in the database;
4. approximate string matching (Levenshtein distance $\leq 2$) with a variation of the entity in the knowledge base. For efficiency, we used a BK-tree (Burkhard and Keller, 1973) for this functionality.

[1] https://github.com/mit-nlp/MITIE

### 2.5 Candidate Selection

#### 2.5.1 Candidate Features

For the selection of the best entity among candidates, we basically rely on two similarity scores.

A first similarity score is based on the similarity between the textual context of the query mention and the textual context of the KB entity. More precisely, it is equal to the cosine similarity between the vectors representing the query document and the text associated with the candidate entity (i.e. its Wikipedia page).

The second similarity score exploits the relations between the entities in the knowledge base. More precisely, we want to determine if the entities appearing in the text around the query mention are linked to the entities in relation with the candidate entity in the knowledge base. We adopted a simple approach to approximate this process (without having to perform the linking of the other entities in the document and to apply slot filling for verifying the actual presence of the relations in the document): for each entity $E$ in the knowledge base, we build in the same vector space as the Wikipedia pages a tf-idf vector containing the list of the entities in relation (either directly or through a CVT) with $E$[2]. We then measure the relation similarity of the candidate entity by the cosine between this vector and the vector representing the query document.

#### 2.5.2 Selection

For integrating all our criteria in a flexible way and choosing the best information to use for the selection of the best candidate, we relied on a statistical classifier. We added to the two similarity scores a set of 4 binary features that indicate the origin of the candidate generation (1 to 4 in previous section), with the idea that a candidate generated by direct string equality is stronger than a candidate generated by approximate string matching.

A classifier is then trained to recognize the best entity among the entity candidates, using the training data provided. More precisely, we used a binary classifier that decides, for each (query, candidate) pair if the query mention is an instance of the candidate entity. The positive examples are the instances taken from the training data, the negative examples are wrong candidates generated from the training data. Since the number of candidates generated for each query may be high (between 1 and $460,055$), we limited the number of negative examples to be $X$ times as big as the number of positive examples. In the submitted run, we used $X = 10$ and a Random Forest classifier.

For every query, the classifier produces a probability for each candidate and the candidate with the highest probability can be selected.

#### 2.5.3 Entity Type Filtering

The expected result of the EDL task must include the type of the entity, which must be one of the expected types (PER, LOC, ORG, GPE, FAC). The query analysis step includes the use of the MITIE tool to extract named entity but the model we used only recognize the types PER, LOC, ORG. Moreover, we did not want to rely only on the quality of the named entity extractor for the entity type: we decided to generate the candidate entities without any constraint on their type. An additional filtering on the entity type is then added during the candidate selection process. More precisely, we keep the 5 best results returned by the classifier, we filter out the candidate entities that do not have a type compatible with the expected types and we select the remaining candidate with the highest score (if any). Simple ad-hoc rules have been used for the compatibility of the entity type found in the knowledge base and the expected types (e.g. 'administrative_division' or 'country' are possible Freebase types for GPE).

A query is finally marked as **NIL** if no candidate entity is found during the candidate generation step or if the classifier or the entity type filtering rejects all candidates.

## 3 Evaluation

### 3.1 Tests on DBpedia KB

Since this is the first time Freebase is used as a knowledge base in the TAC Entity Linking task, we first developed our system using the DBpedia database that was used in previous years (2009 to 2013). Table 1 presents some statistics on the queries for these datasets. In particular, the candidates recall, defined by the percentage of non-NIL queries for which the expected candidate is in the candidate list, seems quite good, for simple candidate generation strategies, and the number of candidates per query is also reasonable (the maximum

[2]In practice, this is equivalent to build a pseudo-document containing the entities linked to $E$ and to process it similarly to the textual descriptions attached to the entities coming from Wikipedia.

Table 1: Candidate statistics for the DBpedia datasets (TAC 2009 to 2013).

| | Nb queries | NIL queries | Nb candidates | NIL cand. | Avg. cand. | cand. recall |
|---|---|---|---|---|---|---|
| 2009 | 3,904 | 2,229 | 208,060 | 949 | 70.41 | 84.0% |
| 2010 | 2,250 | 2,230 | 232,672 | 601 | 141.10 | 89.4% |
| 2011 | 2,250 | 1,126 | 329,508 | 388 | 176.96 | 87.9% |
| 2012 | 2,226 | 1,049 | 420,179 | 117 | 199.23 | 92.4% |
| 2013 | 2,190 | 1,007 | 394,217 | 395 | 219.62 | 83.5% |

Table 2: Entity Linking results on the DBPedia datasets, tested on one year and using all other years for training.

| | strong_all_match | | | recall(strong_link_match) | | |
|---|---|---|---|---|---|---|
| | adaboost | svm_linear | random_forest | adaboost | svm_linear | random_forest |
| 2009 | 77.4% | 74.3% | 70.8% | 65.5% | 65.9% | 70.1% |
| 2010 | 77.1% | 80.4% | 71.1% | 73.6% | 72.5% | 70.4% |
| 2011 | 71.9% | 72.6% | 61.1% | 58.6% | 58.9% | 58.2% |
| 2012 | 51.8% | 50.4% | 49.7% | 46.6% | 48.0% | 47.3% |
| 2013 | 73.8% | 74.1% | 68.8% | 65.8% | 67.1% | 65.5% |

number of candidates per query is between 2,718 and 9,964 depending on the years).

Table 2 presents the results obtained by our system with different classifiers: Random Forest, linear SVM and Adaboost (we rely on the scikit-learn implementation of these classifiers, with no particular optimization of the parameters). We use as evaluation measures *strong_all_match* and *recall(strong_link_match)*, that correspond respectively to the overall accuracy and the KB accuracy in the previous TAC EDL tracks. The results obtained with the proposed method are quite good, even if some datasets seem more difficult than others, such as the 2012 dataset (even if it is the year for which the candidate recall is the higher, the entities also seem to have more ambiguity – more candidates). The differences between the classifiers are not obvious: Adaboost and linear SVM generally perform better on the overall measure, whereas Random Forest can be better on the *strong_link_match* measure (on the non NIL entities).

### 3.2 Tests on Freebase KB

In the TAC 2015 EDL track, both the knowledge base and the number of queries (in the training data provided and in the test data) are much larger. Table 3 presents the candidate statistics on the training and test data. For the test data, we restricted the queries to English queries with named entity mentions (NAM). We also removed from these results the queries with entity type *TTL* that are ignored in the gold standard by the official evaluation program (these queries were in our submitted run). We can see in this table that the same candidate generation strategies generate many more candidates than in the DBPedia case (which can simply be explained by the size of the database), and the candidate recall is also lower.

We present in Table 4 our results, as computed by the official evaluation script. These results are different from the official results (in gray) because they were computed on the gold standard restricted to the set of queries that we actually considered. The evaluation is then performed on the 13,587 remaining queries. Since we do not perform named entity recognition nor focus on the entity types, the results presented are restricted to *strong_nil_match*, *strong_link_match* and *strong_all_match*. We generally achieve a good score of almost 60% f-score results, but we can see that our system tends to produce too many NIL answers (precision on NIL answer is only 49%). We analyze more precisely the errors of our system in the next section.

Table 5 gives a more global view of our results compared to the results of the other participants to the EDL task for the English queries as they are presented in (Ji et al., 2015). The evaluation measures are the *strong_typed_** measures instead of the *strong_** measures of Table 4. This explains that our results (in bold) in this table are lower than our re-

Table 3: Candidate statistics for the TAC 2015 EDL datasets.

| | Nb queries | NIL queries | Nb candidates | NIL cand. | Avg. cand. | cand. recall |
|---|---|---|---|---|---|---|
| training | 12,175 | 3,215 | 5,844,592 | 1,282 | 458.08 | 76.0% |
| test | 13,587 | 3,379 | 6,141,369 | 1,255 | 480.32 | 77.6% |

Table 4: Results obtained on the EDL 2015 English queries (restricted/official).

| ptp | fp | rtp | fn | precis. | recall | f-score | measure |
|---|---|---|---|---|---|---|---|
| 5,464 | 2,810 | 5,464 | 4,744 | 0.660 | 0.535 | 0.591 | strong_link_match |
| 5,464 | 3,104 | 5,464 | 5,399 | 0.638 | 0.503 | 0.562 | strong_link_match |
| 2,592 | 2,721 | 2,592 | 787 | 0.488 | 0.767 | 0.596 | strong_nil_match |
| 2,592 | 2,881 | 2,592 | 1,736 | 0.474 | 0.599 | 0.529 | strong_nil_match |
| 8,056 | 5,531 | 8,056 | 5,531 | 0.593 | 0.593 | 0.593 | strong_all_match |
| 8,056 | 5,985 | 8,056 | 7,135 | 0.574 | 0.530 | 0.551 | strong_all_match |

sults in Table 4, as we didn't focus on determining the type of the entities[3]. According to the decreasing values of the *strong_typed_all_match* measure, we can distinguish three main groups of systems: a first group whose f-score is above 0.65, a second group (with gray background) with f-score values between 0.38 and 0.47 and a last group with f-score values lower than 0.27. Our system is the first system of the middle group. Its situation is quite similar according to the *strong_typed_nil* measure[4]. We will look more closely in the next section at the sources of errors in our system but we can already see from Table 5 that the first way to improve it in the framework of the TAC-EDL evaluation would be to determine the type of the NIL entities.

### 3.3 Error Analysis and Further Results

#### 3.3.1 Missing Entities

We present in Table 6 the number of expected entities that are missing at different steps of the entity linking process. We lose most of the entities during the first step of candidate generation: a more precise analysis show that a few expected entities (8) are missing in our database, i.e. were filtered out when we built our database from the Freebase dump. Examples of other frequent missing entities are acronyms that are not explicitly expanded in the textual context of the query mention (for instance *U.S.* is not a known alias of *United States of America* in Freebase). For these ones, an additional step should be added to the candidate generation process for specifically verifying if any variant of the candidate entities has the initials of the query acronym. In the same spirit, we miss all the nationality adjectives (*French*, *Chinese*, *American* etc.): additional linguistic resources should be used to treat this problem.

| step | nb missing entities |
|---|---|
| candidate generation | 2,287 |
| 5-best selection | 1,325 |
| type filtering | 213 |
| 1-best selection | 967 |

Table 6: Number of correct entities lost at each step of the process.

The entities lost during the filtering on the entity type indicates that either the rules we considered to match Freebase entity types to the expected type are not sufficient or that some entity type information is missing in Freebase.

The 5-best selection and the 1-best selection highly depend on the type and the parameters of the classifier used. It turns out that the chosen classifier (Random Forest) does not perform the best for this task. If we consider the three classifiers used on the DBpedia corpus, the Adaboost classifier tends to give better results: concerning the missing entities, the total number of correct entities lost in both steps

[3]More precisely, we used the type of the selected entity when it existed and assigned a default type (PER) to all NIL entities.

[4]Only one system with a low f-score for the *strong_typed_all_match* has a good f-score for the *strong_typed_nil_match* measure.

Table 5: Official results for all participants for EDL 2015 English queries.

| strong_typed_all_match | | | strong_typed_nil_match | | |
|---|---|---|---|---|---|
| precis. | recall | f-score | precis. | recall | f-score |
| 0.736 | 0.738 | 0.737 | 0.736 | 0.75 | 0.743 |
| 0.661 | 0.681 | 0.671 | 0.764 | 0.648 | 0.701 |
| 0.692 | 0.635 | 0.662 | 0.769 | 0.58 | 0.661 |
| 0.679 | 0.627 | 0.652 | 0.582 | 0.641 | 0.61 |
| **0.486** | **0.449** | **0.467** | **0.424** | **0.536** | **0.473** |
| 0.462 | 0.427 | 0.444 | 0.329 | 0.666 | 0.441 |
| 0.405 | 0.375 | 0.389 | 0.291 | 0.612 | 0.394 |
| 0.272 | 0.252 | 0.262 | 0.758 | 0.515 | 0.614 |
| 0.271 | 0.122 | 0.169 | 0.63 | 0.293 | 0.399 |
| 0.032 | 0.03 | 0.031 | 0.039 | 0.075 | 0.051 |

Table 7: Results obtained on the EDL 2015 English queries, using Adaboost classifier.

| ptp | fp | rtp | fn | precis. | recall | f-score | measure |
|---|---|---|---|---|---|---|---|
| 6,018 | 2,632 | 6,018 | 4,190 | 0.696 | 0.590 | 0.638 | strong_link_match |
| 2,638 | 2,299 | 2,638 | 741 | 0.534 | 0.781 | 0.634 | strong_nil_match |
| 8,656 | 4,931 | 8,656 | 4,931 | 0.637 | 0.637 | 0.637 | strong_all_match |

is decreased by 26% with Adaboost (1,693, compared with 2,292 with Random Forest). An additional optimization of the parameters of the classifier (e.g. for Adaboost, the number of estimators and the learning rate) should also improve the result. We report in Table 7 the results obtained (with the same candidates) using the Adaboost classifier, which allows a 4% improvement on the f-score measures.

### 3.3.2 Impact of Relation Similarities

In our participation, we wanted to test whether we could improve the entity linking process using the relations between entities in the knowledge base. In particular, the entities in Freebase are not restricted to the entities from Wikipedia and all entities do not have a textual description or a page attached to them. We present in Table 8 the results obtained with and without the relation similarity score, both for the submitted result (using Random Forest classifier) and for the result using Adaboost, that gives better results. The results show that the addition of the relation similarity tend to increase significantly the precision on the non-NIL entities for both classifiers but with a significant decrease of the recall score. On the NIL entities, the scores tend to be better without the relations. The overall scores are comparable (note that since both classifier methods use random elements, the results should be averaged on several runs in order to have stronger comparisons).

As stated above, one of the motivation to use the KB relations is to deal with entities that do not have an associated textual description. We present in Table 9 the number of correct entities (according to the gold standard), with respect to the values of their similarity scores on the textual context and on the relations. We see, in this table, that less than 0.5% of the entities have a 0 value for both similarities. 4.5% of the entities have a 0 value for the textual similarity (either because the candidate entity has no textual description or because its intersection with the query document is empty) but has a non-zero value

Table 9: Number of correct entities according to the values of the similarity features: sim(doc) is the cosine similarity between the textual contexts and sim(rel) the cosine similarity between the relation contexts.

| similarities | nb. entities |
|---|---|
| sim(doc)$\neq$0 and sim(rel)$\neq$0 | 6,967 |
| sim(doc)$\neq$0 and sim(rel)=0 | 580 |
| sim(doc)=0 and sim(rel)$\neq$0 | 357 |
| sim(doc)=0 and sim(rel)=0 | 38 |

Table 8: Results on the EDL 2015 English queries, with or without relations.

| | Random Forest | | | | | |
|---|---|---|---|---|---|---|
| | with relations | | | without relations | | |
| measure | precis. | recall | f-score | precis. | recall | f-score |
| strong_link_match | 0.660 | 0.535 | 0.591 | 0.546 | 0.616 | 0.579 |
| strong_nil_match | 0.488 | 0.767 | 0.596 | 0.523 | 0.703 | 0.600 |
| strong_all_match | 0.593 | 0.593 | 0.593 | 0.585 | 0.585 | 0.585 |
| | Adaboost | | | | | |
| | with relations | | | without relations | | |
| measure | precis. | recall | f-score | precis. | recall | f-score |
| strong_link_match | 0.696 | 0.590 | 0.638 | 0.588 | 0.696 | 0.637 |
| strong_nil_match | 0.534 | 0.781 | 0.634 | 0.545 | 0.801 | 0.649 |
| strong_all_match | 0.637 | 0.637 | 0.637 | 0.641 | 0.641 | 0.641 |

for the relation similarity. Among these 357 entities, 193 are kept as final choice by the Random Forest classifier (129 by the Adaboost classifier). Even if these entities could not be found without the relation similarity, they represent a small portion of the query entities and do not have a significant quantitative impact on the overall scores.

The approach we proposed for the relation similarity is relatively simple. This approach could be improved by taking into account more information, both on the query side and on the KB side. Indeed, we considered all words in the query document for this similarity whereas we could consider only the named entities that are in relation with the target mention, in order to have a more symmetric relation representation between the query mention and the candidate. We would need for that a relation extraction tool. Since each entity in relation in the document participates in the disambiguation of the other entities, a collaborative entity linking approach (Chen and Ji, 2011) can be considered. On the other hand, we only considered, in the knowledge base, the entities that are in direct relations with the candidate entity: we could enrich these with more loosely related entities (e.g. entities in relation with related entities). As a naive implementation of this idea would be too costly, a possibility would be to use dense representations of the KB relations that factorize the relation information (Nickel et al., 2012; Chang et al., 2014).

## 4 Conclusion

We have presented in this article the Entity Linking system that we developed for our participation to the TAC EDL 2015 track for the monolingual English diagnostic task. We integrated in our system a score that exploits the relations between the entities in the knowledge base. We show that this measure may increase the precision of the entity linking for non-NIL entities, even if the overall scores are not improved. We plan to extend this relation similarity to take into account the relations between the entities in the query document and extend the relations taken into account in the knowledge base. Since the approach we adopted is not strongly language specific, we also plan to test the same approach for the trilingual task (in the query analysis, the named entity recognition does depend on the language; so we should switch to another tool, such as the Stanford NER tool, that deals with Spanish and Chinese).

## References

W. A. Burkhard and R. M. Keller. 1973. Some Approaches to Best-match File Searching. *Communications of the ACM*, 16(4):230–236, April.

Kai-Wei Chang, Wen tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*. Association for Computational Linguistics, October.

Zheng Chen and Heng Ji. 2011. Collaborative ranking: A case study on entity linking. In *Proceedings of the 2011 Conference on Empirical Methods in Nat-*

*ural Language Processing (EMNLP'11)*, pages 771–781. Association for Computational Linguistics.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*, pages 277–285. Association for Computational Linguistics.

Heng Ji, Joel Nothman, and Ben Hachey. 2014. Overview of TAC-KBP2014 Entity Discovery and Linking Tasks. In *Text Analysis Conference (TAC 2014)*, Gaithersburg, Maryland, USA. National Institute of Standards and Technology.

Heng Ji, Joel Nothman, Ben Hachey, and Radu Florian. 2015. Overview of tac-kbp2015 tri-lingual entity discovery and linking. In *Text Analysis Conference (TAC 2015)*, Gaithersburg, Maryland, USA. National Institute of Standards and Technology.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2012. Factorizing yago: Scalable machine learning for linked data. In *Proceedings of the 21st International Conference on World Wide Web (WWW'12)*, pages 271–280. ACM.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.