Combining Word and Entity Embeddings for Entity Linking

Jose G. Moreno[1], Romaric Besançon[2], Romain Beaumont[3], Eva D'hondt[3], Anne-Laure Ligozat[3,4], Sophie Rosset[3], Xavier Tannier[3,5], and Brigitte Grau[3,4]
firstname.lastname@{irit.fr[1],cea.fr[2],limsi.fr[3]}

[1] Université Paul Sabatier, IRIT, 118 Route de Narbonne, F-31062 Toulouse, France
[2] CEA, LIST, Vision and Content Engineering Lab., F-91191 Gif-sur-Yvette, France
[3] LIMSI, CNRS, F-91405 Orsay, France, Université Paris-Saclay
[4] ENSIIE
[5] Univ. Paris-Sud

Abstract. The correct identification of the link between an entity mention in a text and a known entity in a large knowledge base is important in information retrieval or information extraction. The general approach for this task is to generate, for a given mention, a set of candidate entities from the base and, in a second step, determine which is the best one. This paper proposes a novel method for the second step which is based on the joint learning of embeddings for the words in the text and the entities in the knowledge base. By learning these embeddings in the same space we arrive at a more conceptually grounded model that can be used for candidate selection based on the surrounding context. The relative improvement of this approach is experimentally validated on a recent benchmark corpus from the TAC-EDL 2015 evaluation campaign.

Keywords: Entity Linking, Linked Data, Natural Language Processing and Information Retrieval

1 Introduction

In this paper, we investigate a new approach to candidate selection in the context of the Entity Disambiguation (or Entity Linking) task. This task consists of connecting an entity mention that has been identified in a text to one of the known entities in a knowledge base [16, 25], in order to provide a unique normalization of the mention. Entity Linking sometimes figures as part of a more general framework which globally disambiguates all the concepts in a document with respect to a knowledge base (KB), whether they are named entities or nominal expressions (e.g. Wikify [17] or Babelfy [21]).

An Entity Disambiguation system usually consists of three main steps [11]. First, it analyzes an input (a text) to identify "entity mentions" that need to be linked to the knowledge base; then, for each mention, the system generates several candidate entities from the knowledge base; finally, it selects the best entity among the candidates. One of the main challenges is the extremely large number

of entities present in the knowledge base, and consequently their disambiguation, given that a same mention can refer to different entities, and the correct reference can only be deduced from its surrounding context in a text. Consider the following example: "As soon as he landed at the Bangkok airport, Koirala saw Modi's tweets on the quake, Nepal's Minister ...". The mention "Koirala" is fairly ambiguous, it could refer to "Manisha Koirala", a Nepalese actrice, the "Koirale family", a dominating family in Nepalese politics, "Saradha Koirala", a poet of Nepalese descent or "Sushil Koirala", the Nepalese Prime Minister between 2014 and 2015. In this setting even the context word 'Nepal' will not be of much use, and a disambiguation module must use the information contained within the context to its fullest to arrive at a correct disambiguation. An (accurate) Entity Linking system will map the forms "Koirala" to Wikipedia entity "Sushil Koirala", "Modi" to "Narendra Modi" and "Nepal" to the country entity.

The main contribution of this paper focuses on the last step in this process, i.e. 'candidate selection'. Most of the current approaches to this problem are 'word-based' and consider the words as atomic units when using information on the mention and its context to select the best candidate. We propose a more semantically-oriented approach which is based on the joint learning of word and entity embeddings in the same embedding space. The advantages of learning these representations simultaneously are threefold: (1) The resulting word embeddings are more conceptually grounded as their context (during training) may contain concept vectors which supersede surface variations; (2) Entity embeddings are learned over a large text corpus and attain higher frequencies in training than embeddings that are learned directly over knowledge bases; (3) Since the representations are learned in the same space, we can use a simple similarity measure to calculate the distance between an entity mention and its context (words), and its possible entry (entity) in the KB. In this paper, we focus our efforts on entities that exist in Wikipedia as this is one of the few publicly-available, general purpose corpora that contains both a large amount of text, and is annotated with common entities.

In this paper, we present the following contributions:

- Our EAT model which jointly learns word and entity embeddings (Section 3).
- A global Entity Linking pipeline, integrating this EAT model (Section 4); Note that this model can be integrated as a feature to any kind of supervised approach for Entity Linking.
- An evaluation of this approach using the TAC 2015 "Entity Discovery and Linking" challenge (Section 5). Our result for the task ($P(all) = 0.742$) outperforms a non-EAT baseline and achieves comparable results against the top participants in the challenge.

2 Related Work

Entity Linking approaches are often distinguished by the degree of supervision they require, namely into unsupervised, supervised or semi-supervised methods. The unsupervised methods, proposed for instance in [6] and [9], usually rely only

on some similarity measure between the mention in the text and the entities in the KB. They have the advantage of being simple and easy to implement. However, they also have low performance compared to the supervised methods as was shown in past evaluation campaigns [5]. These supervised methods generally rely on binary classifiers [14, 26] or ranking models [24, 4] for entity disambiguation.

A lot of recent work in this domain has focused on the similarity calculation between the pieces of text (mentions) and entities. A system such as Babelfy [21] manages to connect structured knowledge like WordNet[19], Babelnet[22], and YAGO2[10], through the use of random walks with restart over the semantic networks which jointly represent the whole elements. As a result, a signature is obtained for each element and used to calculate similarities between the different sources, i.e., similarities of elements from WordNet against elements from YAGO2. However, recently word representation techniques have shown surprisingly good results for many NLP tasks [18]. Word embeddings are unsupervised strategies based on observation of text regularities in huge text collections to learn reduced vectors which perform better than the traditional count-based representations [1]. The use of embeddings for existing semantic networks has previously been studied by [2]. Representing knowledge information with embeddings consists in transforming each piece of information of the KB –usually represented by triples $(head, relationship, tail)$– into low dimensional vectors. This transformation is obtained by the optimization of a function that gives high scores when the triples are present in the KB, and low scores otherwise. Based on the work of [2], [27] defines a three components function in charge of the optimization of the word embeddings, the knowledge embeddings and their alignment. This technique manages to mix the knowledge and text, which results in a unique representation space for words, entities and relations. These works are interested in the task of knowledge base completion and only few of them are directly related to the task of Entity Linking [23, 8, 28].

An extension of [27] has been developed in parallel by [8] and [28]. They applied their respective model to several Entity Linking collections. However, like for [27], these works do not directly use the context of an entity mention to built the vector representation but use an alignment function to achieve some matching between the mention and the entity. [23] prefers to use the document representation of [13] to jointly represent Wikipedia pages and words. In both cases the joint space of entities and words is used to calculate similarities between them. [29] and their later work [30] integrated entity embeddings within their Entity Linking system. However, these entity embeddings were learned over concatenated documents with only sequences of entities (where entities are ordered as they are found in annotated documents or by following short KB paths) followed by the text content to align the word and entity representations.

3 Combining Word and Entity Embeddings

Learning representations of words and entities simultaneously is an interesting approach to the problem of Entity Linking, as it allows for an even larger degree

of normalization than the regular grouping of word embeddings (i.e. of words that have similar or related meanings) in the vector space already provides. While previous approaches indirectly addressed the problem by learning separate representations and aligning them [8, 28] or by concatenating entity-only with text-only sequences [29, 30], we opt to learn them simultaneously. In this section, we present a model that is able to combine word and entity embeddings by only using their context. As a consequence, our model can be considered as an extension of the initial words embedding model [18] or its variation [13].

3.1 Definitions

A corpus is a sequence of words w or anchor texts a, where the words belong to the vocabulary V and an anchor text $a = (w, e)$ is a tuple consisting of a word $w \in (V \cup \emptyset)$, and an entity $e \in \xi$, where ξ is the collection of entities in the KB. In all cases, the bold letters correspond to the respective vectors $\boldsymbol{e}$,$\boldsymbol{w} \in \mathbb{R}^d$, where $\mathbb{R}^d$ is called the combined space and d defines the number of dimensions.

3.2 Extended Anchor Text

To obtain $\boldsymbol{w}$ and $\boldsymbol{e}$ in the same space, we introduce the concept of Extended Anchor Text (EAT). Through this, we combine entity information with its anchor text, and consequently introduce it into the corpus. To obtain EATs, the mention of an anchor text a_i is redefined as $a'_{ij} = (w'_i, e_j)$ where $w'_i = w_i$ if w_i is not empty, otherwise w'_i is equal to the set of words present in e_j. For an illustration of the decomposition into anchor text and entity, see Figure 1. We redefine a corpus like a sequence of EATs a', so the full vocabulary is defined by $\mathbb{F} = \{V \cup \xi\}$, that is, the set of embeddings to be learned now contains both words (including mentions) and entities from the knowledge base.

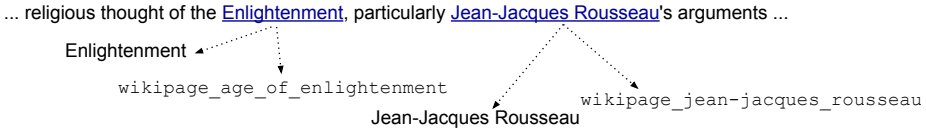

Fig. 1. Illustration of mention-entity decomposition in the EAT model

3.3 The EAT Model

The objective of our model is to find a representation for each element in $\mathbb{F}$ based on the surrounding words/EATs in a sentence. Similarly to [18], we define the probability between two elements in the corpus as in Equation 1.

$$p(c_o|c_i) = \sum_{f_o \in c_o} \sum_{f_i \in c_i} \frac{exp(\boldsymbol{f_o}^T \boldsymbol{f_i})}{\sum_{j=1}^{|\mathbb{F}|} exp(\boldsymbol{f_j}^T \boldsymbol{f_i})} \tag{1}$$

where the elements in $\mathbb{F}$ are identified as f that can represent either a word or an entity, and the words or EATs in a corpus are identified as c. Note that if c_o and c_i are words, the Equation 1 becomes the softmax function into two words and the double sum disappears. The optimization process consists of maximizing the average log probability defined in Equation 1 over a corpus composed by EATs.

The implementation of the EAT models does not imply big changes in actual versions of the original model [18] such as can be found in Word2Vec[6]. As Equation 1 is equivalent to the originally proposed by [18] when c is a word, we just need to adapt it for the case when c is an EAT. The adaptation consists in the expansion of c into their possible combinations but keeping the context static, e.g., the context is the same for the word and the entity. Similarly, if c is part of the context, it is the context that must be expanded. Figure 2 shows the expansion that occurs when the word vector and entity vector that are linked to the entity mention 'Enlightenment' are trained during the training phase.

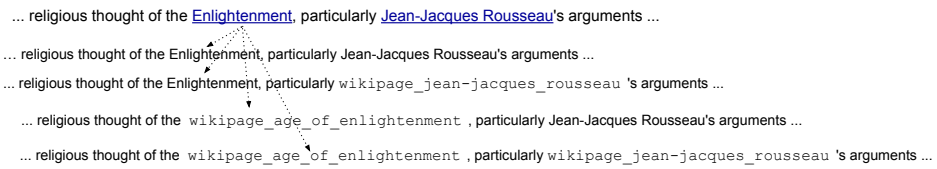

Fig. 2. Illustration of the expanded training contexts for the different embedding types in the EAT model. Please note that only entity mentions are decomposed, and consequently have multiple training moments, i.e. one pass for the anchor text and a separate pass for the entity embedding.

The main advantages of our model are:

- actual methods and embeddings based on the original method proposed by [18] are directly usable within our model including the skip-gram as well as the CBOW configuration;
- anchor texts in publicly available corpus are taken into account within the model allowing us to represent words and entities in a unique space;
- vectors for words and entities are learned using their context instead of indirect relations between them (like the alignment strategy used by [8, 28]), which is more similar to distant supervision techniques such as [20].

4 Entity Linking system

Our Entity Linking system relies on a standard architecture [11] composed of two main steps: for a given entity mention and its textual context, a first module generates possible candidate entities for the linking and a second one takes as input the different candidate entities and selects the best one.

[6] Several open source implementations are available online. We have used Gensim and Hyperwords, available at `https://radimrehurek.com/gensim/models/word2vec.html` and `https://bitbucket.org/omerlevy/hyperwords` respectively.

4.1 Generation of Candidate Entities

The generation of the candidate entities relies on the analysis of the entity mention and its textual context. In this study, we focus mainly on the disambiguation of entities, not their recognition. Therefore, we consider that the offsets of the entity mentions to disambiguate are given as input to the system. A complementary recognition step of entity mentions in the text is nonetheless carried out, in order to associate a type (Person, Location, Organization) with the entity mentions[7] and define their context in terms of surrounding entities (we consider only the explicitly named entity mentions and we ignore the nominal and pronominal mentions). As the entity mention is used to retrieve candidates from the KB, two expansion heuristics are proposed to include variations of the target entity mention. Both of them can be considered as simple co-reference approaches within a document: (i) if the entity mention is an acronym, we search for entity mentions of the same type whose initials match the acronym (ii) we search for entity mentions who include the target entity mention as a substring. The retrieved entity mentions are added as variations for the original entity mention and used to increase the candidates set.

After the analysis of the entity mention, candidate entities are generated by comparing the entity mention (and its variations) to the names of the entities of the KB [7]. We use the following strategies :

- Equality between the forms of the entity mention and an entity in the KB;
- Equality between the form of the entity mention and a variation (alias or translation) of an entity in the KB;
- Inclusion of the form of the entity mention in the name or one of the forms of the variations of an entity in the KB;
- String similarity between the form of the entity mention and a variation of an entity in the KB. We use the Levenshtein distance, which is well suited to overcome the spelling errors and name variations. In the experiments, we considered an entity in the KB as a candidate entity if its form or any of its variations have a distance with the form of the entity mention ≤ 2. For better efficiency, we exploited a BK-tree structure [3] for this selection.
- Information Retrieval model: an information retrieval model is used to index all the known forms of the entities in the KB as documents: We can then select all close variants, weighted by their tf-idf, to find suitable candidates. Lucene was used as search engine.

The candidate entities are also filtered in order to keep only entities that have at least one of the expected entity types (e.g. Person, Location, Organization).

4.2 Selection of the Best Candidate Entity

The objective of this step is to find the correct candidate entity from the set of generated candidate entities. To this purpose, a classifier is trained to recognize the best entity among the entity candidates, using training data on dis-

[7] Named entities were recognized using MITIE https://github.com/mit-nlp/MITIE .

ambiguated entity mentions. More precisely, each candidate entity is associated with a set of features:

- a set of features associated with the strategy that was used for the generation of this candidate entity: binary features for the simple matching strategies, as well as the value of the similarity score for the Information Retrieval model;
- two similarity scores accounting for a global context obtained by comparing the textual context of the entity mention with a textual description of the entities in the KB;
- one score accounting for global popularity of the entity obtained by counting the number of inlinks present in Wikipedia and applying a log normalization;
- four similarity scores are added based on the EAT embeddings, that account for a narrower context. As this context is made of few words, using embeddings allows to overcome the problem of lexical gaps.

Textual similarity scores For an entity mention q and a possible entity candidate e from the KB, we consider three vectors representing three textual contexts: the document in which q appears, noted $d(q)$, the Wikipedia page associated with e, noted $w(e)$ and a text combining the set of entities that are in relation with e in the KB, noted $r(e)$. Each text is represented by a vector using a standard Vector Space Model, with a *tf-idf* weighting scheme, $d(q) = (d_1, \ldots, d_n)$ with $d_i = tf(t_i, d) \times idf(t_i)$, where $tf(t_i, d)$ is a function of the frequency of the term t_i in the document d and $idf(t_i)$ is an inverse function of the document frequency of the term in the collection. All representations are built in a common vector space, constructed from a full Wikipedia dump (the *idf* scores are therefore computed on this complete collection of documents). The scores are then the cosine similarities between the context vector of the entity mention and each of the context vectors of the entity from the KB:

$$sim_d(q, e) = cos(d(q), w(e)) = \frac{\sum_i d_i.w_i}{||d(q)||.||w(e)||}$$

$$sim_r(q, e) = cos(d(q), r(e)) = \frac{\sum_i d_i.r_i}{||d(q)||.||r(e)||}$$

Similarity scores based on EAT embeddings From the document $d(q)$, the paragraph $p(q)$ where the mention q occurs is extracted. Using the offsets provided in the data set, we extract the previous, current and next sentence to where the mention was found to built $p(q)$. Then, the average value of cosine similarities between each word from paragraph and the entity is calculated (EAT_1). The cosine similarity is calculated between the average vector from the paragraph and the entity (EAT_2). The average of the top-k (EAT_3) similarities is used as feature[8]. Finally, the cosine similarity between the entity mention and the entity is added (EAT_4). Equations for the four features are defined below.

$$EAT_1(e, p(q)) = \frac{\sum_{w_i \in p(q)} cos(\boldsymbol{e}, \boldsymbol{w}_i)}{||p(q)||} \qquad EAT_2(e, p(q)) = cos(\boldsymbol{e}, \frac{\sum_{w_i \in p(q)} \boldsymbol{w}_i}{||p(q)||})$$

[8] In our experiments k was fixed to 3.

$$EAT_3(e, p(q)) = \frac{\sum_{i=1\ldots k} argmax_{w_i \in p(q)} cos(\boldsymbol{e}, \boldsymbol{w}_i)}{k} \qquad EAT_4(e, w_m) = cos(\boldsymbol{e}, \boldsymbol{w}_m)$$

where $argmax_{w_i} cos(\boldsymbol{e}, \boldsymbol{w}_i)$ returns the i-th most similar word, in terms of cosine similarity, to $p(q)$.

Classifier trained for candidate selection We then train a binary classifier that associates the given set of features with a decision whether the candidate entity is the correct one for the entity mention. Using the training data, we generate the candidate entities from the entity mentions. The positive examples for the training are then formed by the (entity mention, candidate) pairs that correspond to the expected link in the reference. The negative examples are pairs with wrong candidates generated for the entity mentions. Since the number of candidates generated for each mention may be very high (between 1 and 460 in our experiments), the positive and negative classes are very imbalanced. We deal with this problem by using undersampling, limiting the number of negative examples to be 10 times the number of positive examples[9]. Each decision of the classifier is then weighted by the probability estimate of the classifier and the candidate entity with the highest probability is selected as the final disambiguated entity. In the standard entity disambiguation task, the system must also be capable of determining when an entity mention does not link to any entity in the KB (these are referred as NIL entities). In our approach, this occurs if no candidate is generated or if all candidates are rejected by the classifier.

Due to the particular nature of the feature vectors which combines dimensions of a very different nature such as binary features versus floats, we tested several classifiers. Models such as Adaboost, Random Forests, Decision Trees and SVM models (linear and RBF kernels) were tested. Combining Adaboost with Decision Trees as base estimator turned out to be the best classifier on the training data (using the non-EATs features and a 10-fold cross validation schema). Further results are obtained using this classifier.

5 Experiments and results

5.1 Learning the embeddings

In order to apply the EAT model, a collection of documents where entities are referenced within the text is needed. This is the case for Wikipedia[10] where each page is considered as an entity. The anchor texts were slightly modified to indicate the part that corresponds to a word and the part that corresponds to an entity. A mapping to DBpedia is constructed for the entity and a prefix allows us to identify the entities. Next section presents our preliminary results of the implementation used for our model, based on *Gensim* or *Hyperwords*.

[9] We tested several values for this ratio between positive/negative sample on the training data and kept the value that achieved the best result.

[10] We used the data dump available in June 2016.

5.2 Evaluation of the embeddings

In our first experiments we want to evaluate the quality of the learned embeddings by testing on the well-known analogy data set[18]. The analogy task goes as follows: Given a pair of words between which a relation holds, e.g. 'Paris' - 'France', predict the empty slot for a new pair such as 'Rome' - <?> . (Spoiler: It's 'Italy'.) The original analogy data set consists of syntactic and semantic word pairs. Our experiments were focused on the semantic relations. As our intention is to evaluate the quality of the obtained vectors for words and entities, each example was mapped to their string equivalent entities. This process was possible only for four of the five original semantic relations due the missing entities for the family relations in Wikipedia. Note that the remaining four semantic relations deal only with locations or currencies.

Early experiments were performed with the *Hyperwords* tool used to implement the EAT model. First, we used the suggested configuration by the authors of this tool [15] (skip-gram configuration, $negative\_sampling = 5$, $window = 2$, $vectors = words + context$). Results fairly approximate the values previously reported by [15] and outperform values reported by [18]. The EAT model performs slightly worse than the results obtained by the original *Hyperwords* implementation. The smallest difference is up to 1.2% when the addition function is used (61.9%) and the largest is up to 2.8% when the multiplication function is used (67.6%)[11]. This difference between the basic model and the EAT version is due to the additional points (the extra entities) that are represented in the space. A similar situation was observed during our experiments, e.g., lower performances are obtained when the vocabulary size is increased by the modification of the threshold frequency (frequency values under the threshold are filtered out of the training corpus). As mentioned by [15], correct parametrization is a core element to achieve top performances. Indeed, when a high value is used as frequency threshold the results are competitive compared with the state of the art for the task of analogy (see column *EAT-hyperwords* in Table 1), but many entities are missing. The high number of entities filtered out by the threshold highly impacts the performance obtained with the entities. On the other hand, when the threshold frequency is set to a lower value, many entities are represented but the word analogy performance decays, e.g., when parameters are relaxed the performance for words tends to decrease, but more Wikipedia pages are represented by the model. This situation impacts the results when only the entities are used.

Results of individual subgroups are shown in Table 1 for our model using the EAT-*hyperwords* or EAT-*Gensim* implementations. The *words* column reports the results obtained using only words and *entities* column reports the results for their equivalent entity name. Finally, we have reported experiments in which *word* and *entities* are combined in the *entity→words* column. In the later case, we have replaced by their respective word when the entity was not part of the vocabulary. The results clearly outperform the entity-only column and start to approach the strong words-only based results. Indeed, the *entity→words* results

[11] More details about the addition and multiplication functions can be found in [15].

Table 1. Accuracy by semantic subgroup in the analogy task for words/entities using a high value (EAT-*hyperwords*) or low value (EAT-*Gensim*) as frequency threshold.

	EAT-*hyperwords*			EAT-*Gensim*	
Subgroup	words	entities	entity→word	words	entities
capital-com-countries	95.7%	63.0%	87.5%	75.7%	77.5%
capital-world	77.0%	37.3%	81.3%	49.7%	80.0%
currency	8.2%	0.0%	5.2%	0.0%	0.0%
city-in-state	72.3%	25.8%	62.6%	31.7%	89.8%

outperform the words-only results for the *capital-world* subgroup. In column EAT-*Gensim*, it is reported the results of our EAT-*Gensim* implementation with the relaxed parameters (frequency threshold equal to 30 to words and to 0 for entities). Results for entities clearly improve those for words in subgroups *capital-world* and *city-in-state*, but no significant changes are observed in subgroups *capital-common-countries* or *currency*[12]. Indeed, overall results using semantic and syntactic groups for only-words are clearly less performant (40.31%) than our EAT-*hyperwords* implementation (61.9%). However, EAT-*Gensim* is preferred because more entities are represented despite the fact that EAT-*Gensim* has a worse performance when compared with EAT-*hyperwords*.

Further experiments are performed with the EAT relaxed parameters version, e.g., our EAT implementation based on *Gensim* in order to have the maximum number of words and entities represented in the joint space.

5.3 Dataset and evaluation measures for Entity Linking

To validate our approaches on the Entity Linking task, we use the benchmark from the EDL (Entity Discovery and Linking) task of the TAC 2015 evaluation campaign. We only consider the monolingual English Diagnostic Task, where the entity mentions in the query texts are already given as input, since our main focus in this work is on the linking and not the detection of the entity mentions. Table 2 shows the main features of the used data set: the number of documents, the number of entity mentions to disambiguate (the goal of the task is to disambiguate all the entity mentions present in the considered documents), and the number of entity mentions that do not have a corresponding entity in the knowledge base (mentions NIL). The knowledge base used in this campaign is built from Freebase [12]. The whole Freebase snapshot contains $43M$ entities but a filter was applied to remove some entity types that were not relevant to the campaign (such as music, book, medicine and film), which reduced it to $8M$ entities. Among them, only 3,712,852 (46%) have an associated content in Wikipedia and can thus be associated with an embedding representation. In order to improve the candidate generation process, we also enriched the knowledge

[12] Results for family are not calculated due the missing entities in Wikipedia.

Table 2. Description of the dataset used in the evaluation process

	TAC 2015 training	TAC 2015 testing
Nb. docs.	168	167
Nb. mentions	12,175	13,587
Nb. mentions NIL	3,215	3,379

base with new entity expressions automatically extracted from Wikipedia: more precisely, we added all links from disambiguation pages and redirection pages as possible forms of the entities.

For the evaluation scores, we used the standard precision/recall/f-score measures on the correct identification of the KB entity and its type when it exists (*link*), on the correct identification of a NIL mention (*nil*) or the combined score for both cases (*all*). Compared to the official evaluation measures from the campaign, we do not consider the evaluation of the clustering of the NIL mentions referring to the same entity. These measures correspond to the *strong_typed_link_match*, *strong_typed_nil_match* and *strong_typed_all_match* measures from the TAC EDL campaign. Formally, if we note, for an entity mention e, the KB entity e_r associated with e if in the reference, the KB entity e_t associated with it by our system and $N(x)$ the number of entity mentions that verify x, then these measures are defined by:

$$P(nil) = \frac{N(e_t = \text{NIL} \wedge e_r = \text{NIL})}{N(e_t = \text{NIL})} \quad R(nil) = \frac{N(e_t = \text{NIL} \wedge e_r = \text{NIL})}{N(e_r = \text{NIL})}$$

$$P(link) = \frac{N(e_t = e_r \wedge e_t \neq \text{NIL})}{N(e_t \neq \text{NIL})} \quad R(link) = \frac{N(e_t = e_r \wedge e_t \neq \text{NIL})}{N(e_r \neq \text{NIL})}$$

$$P(all) = \frac{N(e_t = e_r)}{N(e_t)}$$

Note that, for the *all* measure, precision, recall and f-score are equal, provided that the system gave an answer for all the entity mentions ($N(e_t) = N(e_r)$).

5.4 Evaluation of candidate entity generation

In this section we discuss the results of the candidate generation. Table 3 presents some statistics on the candidate generation. We denote C the set of candidates, C_{NIL} the set of mentions for which no candidate is proposed, C_{AVG} the average number of candidates per query and Recall(C) the candidate recall, defined by the percentage of non-NIL queries for which the excepted KB entity is present in the set of candidate entities.

Firstly, when considering all strategies for candidate generation, without any filtering, we achieve a high candidate recall, with 95% of the non-NIL entity mentions found among the candidates. We also note that this leads to a large number of candidate entities per mention. When applying a filtering on the

Table 3. Statistics on candidate generation.

	All candidates			
	$\|C\|$	$\|C_{NIL}\|$	C_{AVG}	Recall(C)
training	6,843,513	781	562.1	95.60%
test	8,339,648	499	613.8	94.19%
	Entity Type Filtering			
	$\|C\|$	$\|C_{NIL}\|$	C_{AVG}	Recall(C)
training	3,179,795	952	261.2	92.43%
test	3,810,382	626	280.4	90.36%
	Lucene+Null Simil Filtering			
	$\|C\|$	$\|C_{NIL}\|$	C_{AVG}	Recall(C)
training	1,723,470	952	141.6	90.27%
test	1,921,577	625	141.4	87.95%

entity types, i.e. we keep only the KB entities for which we can derive one of the expected entity types (PER, LOC, ORG, GPE, FAC), we reduce by more than half the number of candidate entities, which give a sounder base for the classifier: even if the recall is decreased (around 90%), the Entity Linking score is improved.

An analysis of the generated candidates also showed that the candidates returned only by Lucene and the candidates for which the similarity scores are both null were not often the right ones: once again, removing these entities before learning the classifier leads to better results, with a global linking f-score of 72.8%, for a candidate recall around 88%. This last strategy is one used in the following results. Further work and analysis on this candidate generation step is needed, in order to determine more sophisticated filtering strategies, that will allow to keep the good candidates without generating too much noise through spurious candidates.

5.5 Entity Linking Results

In Table 4 we present the results obtained for the global Entity Linking task, using the different features proposed in this paper. The *baseline* result is obtained using only the features from the candidate generation and the cosine similarity scores on the textual context. The other results are obtained when adding each of the scores computed with the embeddings with the EAT model. The last column uses all the scores combined. Since our Entity Linking model relies on methods that have some random elements (negative example selection for undersampling and internal sampling from the Adaboost classifier), the results presented are average results on 10 runs.

Table 4. Entity Linking Results with the EAT model.

	baseline	$+EAT_1$	$+EAT_2$	$+EAT_3$	$+EAT_4$	$+EAT_{1/2/3/4}$
P(nil)	0.598	0.604	0.608	0.605	0.605	**0.606**
R(nil)	0.815	0.830	0.825	0.828	0.830	**0.838**
F(nil)	0.690	0.699	0.700	0.700	0.700	**0.704**
P(link)	0.796	0.806	0.800	0.804	0.806	**0.814**
R(link)	0.699	0.706	0.706	0.706	0.707	**0.710**
F(link)	0.745	0.752	0.750	0.752	0.753	**0.759**
P(all)	0.728	0.737	0.735	0.737	0.737	**0.742**

These results show a significant improvement of the scores when using the embeddings of the EAT model, over the baseline. We also note that the improvement obtained with each individual EAT feature is comparable and the combined features give the best results.

When compared to the results from the participants to the TAC-EDL 2015 campaign [12], the best F-score result for the linking task was 0.737, on the *strong_typed_all_match* measure. We therefore achieve with this model better results than the state of the art[13]. A close examination of the results shows some examples of the improvements obtained by using a narrow semantic context through the EAT model. For example, in the ambiguous cases where a person is only referred to using his last name, our model is consistently better in selecting the correct entity. In the phrase "As soon as he landed at the Bangkok airport, Koirala saw Modi's tweets on the quake, Nepal's Minister for Foreign Affairs Mahendra Bahadur Pandey said on Tuesday." the mention "Modi" is correctly identified as "Navendra Modi" instead of "Modi Naturals" an oil processing company based in India. Similar performances were observed for the entity type location. For example, the EAT model correctly identified "Montrouge", as the French town near Paris, instead of the french actor Louis (Émile) Hesnard known as "Montrouge" (who was born in Paris) for the sentence "The other loose guy who killed a cop in montrouge seems to have done the same. And there are report of two other armed men running around in Paris. It's kind of a chaos here."

6 Conclusion

In this paper we presented a model capable of jointly representing word and entities into a unique space, the EAT model. The main advantage of our model is the capability of representing entities as well as word embeddings in context during training. Our model –based on anchor texts– accurately represents the entities in the jointly learned embedding space, even better that the words because entities (Wikipedia pages) are used in contexts which clearly represent their meanings. Indeed, this is the main advantage of our model, the direct use

[13] Our evaluation does not take into account the nominal mentions of entities.

of contexts for the construction of the entities embeddings skipping an extra alignment task previously used by [27, 8, 28] or corpus concatenation [29, 30].

We showed that the EAT model can be integrated into a standard entity linking architecture without any extra effort. Four different features have been proposed to encode similarities between the context and the candidates. For evaluation, we have used a recent and competitive entity linking dataset of the TAC-EDL 2015 campaign. The results show that individual EAT features as well as their combination helps to improve classical similarity metrics. Our final result for the task ($P(all) = 0.742$) outperforms the baseline and hypothetically achieves the first position in the mentioned evaluation campaign.

Acknowledgements

This work was supported by the French National Agency for Research under the grant PULSAR-FUI-18 (PUrchasing Low Signals and Adaptive Recommendation).

References

1. Baroni, M., Dinu, G., Kruszewski, G.: Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proceedings of the 52nd Annual Meeting of the ACL. pp. 238–247 (June 2014)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems 26, pp. 2787–2795 (2013)
3. Burkhard, W.A., Keller, R.M.: Some Approaches to Best-match File Searching. Communications of the ACM 16(4), 230–236 (Apr 1973)
4. Cao, Z., Tao, Q., Tie-Yan, L., Ming-Feng, T., Hang, L.: Learning to Rank: From Pairwise Approach to Listwise Approach. In: 24th International Conference on Machine Learning (ICML 2007). pp. 129–136. Corvalis, Oregon, USA (2007)
5. Cassidy, T., Chen, Z., Artiles, J., Ji, H., Deng, H., Ratinov, L.A., Zheng, J., Han, J., Roth, D.: CUNY-UIUC-SRI TAC-KBP2011 Entity Linking System Description. In: Text Analysis Conference (TAC 2011) (2011)
6. Cucerzan, S.: Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In: 2007 Joint Conference on EMNLP-CoNLL. pp. 708–716 (2007)
7. Dredze, M., McNamee, P., Rao, D., Gerber, A., Finin, T.: Entity Disambiguation for Knowledge Base Population. In: 23rd International Conference on Computational Linguistics (COLING'10). pp. 277–285. Beijing, China (2010)
8. Fang, W., Zhang, J., Wang, D., Chen, Z., Li, M.: Entity disambiguation by knowledge and text jointly embedding. CoNLL 2016 p. 260 (2016)
9. Han, X., Zhao, J.: NLPR_KBP in TAC 2009 KBP Track: A Two-Stage Method to Entity Linking. In: Text Analysis Conference (TAC 2009) (2009)
10. Hoffart, J., Suchanek, F., Berberich, K., Weikum, G.: Yago2: A spatially and temporally enhanced knowledge base from wikipedia. Artif. Intell. 194, 28–61 (2013)
11. Ji, H., Nothman, J., Hachey, B.: Overview of TAC-KBP2014 Entity Discovery and Linking Tasks. In: Text Analysis Conference (TAC 2014) (2014)

12. Ji, H., Nothman, J., Hachey, B., Florian, R.: Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking. In: Text Analysis Conference (TAC 2015) (2015)
13. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. Proceedings of The 31st ICML pp. 1188–1196 (2014)
14. Lehmann, J., Monahan, S., Nezda, L., Jung, A., Shi, Y.: LCC Approaches to Knowledge Base Population at TAC 2010. In: Text Analysis Conference (2010)
15. Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics 3, 211–225 (2015)
16. Ling, X., Singh, S., Weld, D.: Design Challenges for Entity Linking. Transactions of the Association for Computational Linguistics (TACL) 3, 315–328 (2015)
17. Mihalcea, R., Csomai, A.: Wikify! Linking Documents to Encyclopedic Knowledge. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management. pp. 233–242. ACM, Lisbon, Portugal (2007)
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26, pp. 3111–3119 (2013)
19. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM 38(11), 39–41 (1995)
20. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th IJCNLP. pp. 1003–1011. ACL '09 (2009)
21. Moro, A., Raganato, A., Navigli, R.: Entity linking meets word sense disambiguation: a unified approach. Transactions of the Association for Computational Linguistics (TACL) 2, 231–244 (2014)
22. Navigli, R., Ponzetto, S.P.: Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence 193, 217–250 (2012)
23. Pappu, A., Blanco, R., Mehdad, Y., Stent, A., Thadani, K.: Lightweight multilingual entity extraction and linking. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. pp. 365–374. WSDM 17, ACM (2017)
24. Shen, W., Jianyong, W., Ping, L., Min, W.: LINDEN: Linking Named Entities with Knowledge Base via Semantic Knowledge. In: 21st International Conference on World Wide Web (WWW'12). pp. 449–458. Lyon, France (2012)
25. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: Issues, techniques, and solutions. Transactions on Knowledge and Data Engineering (2015)
26. Varma, V., Bharath, V., Kovelamudi, S., Bysani, P., GSK, S., N, K.K., Reddy, K., Kumar, K., Maganti, N.: IIT Hyderabad at TAC 2009. In: Text Analysis Conference (TAC 2009) (2009)
27. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph and text jointly embedding. In: The 2014 Conference on Empirical Methods on Natural Language Processing. ACL – Association for Computational Linguistics (October 2014)
28. Yamada, I., Shindo, H., Takeda, H., Takefuji, Y.: Joint learning of the embedding of words and entities for named entity disambiguation. Proceedings of the 20th SIGNLL CoNLL pp. 250–259 (2016)
29. Zwicklbauer, S., Seifert, C., Granitzer, M.: Doser - A knowledge-base-agnostic framework for entity disambiguation using semantic embeddings. In: 13th E Conference, ESWC 2016. pp. 182–198 (2016)
30. Zwicklbauer, S., Seifert, C., Granitzer, M.: Robust and collective entity disambiguation through semantic embeddings. In: 39th Int ACM Conference on Research and Development in Information Retrieval (SIGIR). pp. 425–434 (2016)