Neural Cross-Lingual Entity Linking

Avirup Sil, Gourab Kundu, Radu Florian, Wael Hamza

IBM Research AI
1101 Kitchawan Road
Yorktown Heights, NY 10598
{avi, gkundu, raduf, whamza}@us.ibm.com

Abstract

A major challenge in Entity Linking (EL) is making effective use of contextual information to disambiguate mentions to Wikipedia that might refer to different entities in different contexts. The problem exacerbates with cross-lingual EL which involves linking mentions written in non-English documents to entries in the English Wikipedia: to compare textual clues across languages we need to compute similarity between textual fragments across languages. In this paper, we propose a neural EL model that trains fine-grained similarities and dissimilarities between the query and candidate document from multiple perspectives, combined with convolution and tensor networks. Further, we show that this English-trained system can be applied, in zero-shot learning, to other languages by making surprisingly effective use of multi-lingual embeddings. The proposed system has strong empirical evidence yielding state-of-the-art results in English as well as cross-lingual: Spanish and Chinese TAC 2015 datasets.

Introduction

Entity Linking (EL) is the task of associating a specific textual mention of an entity (henceforth query entity) in a given document (henceforth query document) with an entry in a large target catalog of entities, often called a knowledge base or KB, and is one of the major tasks in the Knowledge-Base Population (KBP) track at the Text Analysis Conference (TAC) (Ji et al. 2014; 2015). Most of the previous EL research (Cucerzan 2007; Ratinov et al. 2011; Sil and Yates 2013) have used Wikipedia as the target catalog of entities, because of its coverage and frequent updates made by the community of users.

Some ambiguous cases for entity linking require computing fine-grained similarity between the context of the query mention and the title page of the disambiguation candidate. Consider the following examples:

e_1: Alexander Douglas Smith is an American football quarterback for the Kansas City Chiefs of the National Football League (NFL).

e_2: Edwin Alexander "Alex" Smith is an American football tight end who was drafted by the Tampa Bay Buccaneers in the third round of the 2005 NFL Draft.

e_3: Alexander Smith was a Scottish-American professional golfer who played in the late 19th and early 20th century.

q: Last year, while not one of the NFL's very best quarterbacks, Alex Smith did lead the team to a strong 12-4 season.

Here, e_1, e_2 and e_3 refer to the Wikipedia pages of three sportsmen (only first sentence is shown), known as "Alex Smith"; q refers to the sentence for the query mention "Alex Smith". Since words in e_3 belong to a different domain (golf) than q (American football), simple similarity based methods *e.g.* TF-IDF based cosine similarity will have no difficulty in discarding e_3 as disambiguation for q. But words in e_1 and e_2 contain significant overlap (both are American football players) even in key terms like NFL. Since "Alex Smith" in q is a quarterback, correct disambiguation for q is e_1. This requires fine-grained similarity computation between q and the title page of e_1. In this paper, we propose *training* state-of-the-art (SOTA) similarity models between the context of the query mention and the page of the disambiguation candidate from Wikipedia such that the similarity models can learn to correctly resolve such ambiguous cases. We investigate several ways of representing both the similarity and coherence between the query document and candidate Wikipedia pages. For this purpose, we extract contextual information at different levels of granularity using the entity coreference chain, as well as surrounding mentions in the query document, then use a combination of convolutional neural networks (CNN), LSTMs (Hochreiter and Schmidhuber 1997), Lexical Composition and Decomposition (Wang, Mi, and Ittycheriah 2016), Multi-Perspective Context Matching (MPCM) (Wang et al. 2016), and Neural Tensor Networks (Socher et al. 2013a; 2013c) to encode this information and ultimately perform EL.

The TAC community is also interested in cross-lingual EL (Tsai and Roth 2016; Sil and Florian 2016): given a mention in a foreign language document *e.g.* Spanish or Chinese, one has to find its corresponding link in the English Wikipedia. The main motivation of the task is to do Information Extraction (IE) from a foreign language for which we have extremely limited (or possibly even no) linguistic resources and no machine translation technology. The TAC 2017 pilot evaluation [1] targets really low-resource lan-

---

[1] http://nlp.cs.rpi.edu/kbp/2017/taskspec_pilot.pdf

guages like Northern Sotho or Kikuyu which only have about 4000 Wikipedia pages which is a significantly smaller size than the English Wikipedia. Recently, for cross-lingual EL, (Tsai and Roth 2016) proposed a cross-lingual wikifier that uses multi-lingual embeddings. However, their model needs to be re-trained for every new language and hence is not entirely suitable/convenient for the TAC task. We propose a zero shot learning technique (Palatucci et al. 2009; Socher et al. 2013b) for our neural EL model: once trained in English, it is applied for cross-lingual EL without the need for re-training. We also compare three popular multi-lingual embeddings strategies and perform experiments to show which ones work best for the task of zero-shot cross-lingual EL. The results show that our methods not only obtain results that are better than published SOTA results on English, but it can also be applied on cross-lingual EL on Spanish and Chinese standard datasets, also yielding SOTA results.

Entity Linking Formulation

We formalize the problem as follows: we are given a document D in any language, a set of mentions $M_D = m_1; \ldots; m_n$ in D, and the English Wikipedia. For each mention in the document, the goal is to retrieve the English Wikipedia link that the mention refers to. If the corresponding entity or concept does not exist in the English Wikipedia, "NIL" should be the answer.

Given a mention $m \in M_D$, the first step is to generate a set of link candidates L_m. The goal of this step is to use a *fast match* procedure to obtain a list of links which hopefully include the correct answer. We only look at the surface form of the mention in this step, and use no contextual information. The second essential step is the ranking step where we calculate a score for each title candidate $l_j^{(m)} \in L_m$, which indicates how relevant it is to the given mention. We represent the mention using various contextual clues and compute several similarity scores between the mention and the English title candidates based on multilingual word and title embeddings. A ranking model learned from Wikipedia documents is used to combine these similarity scores and output the final score for each candidate. We then select the candidate with the highest score as the answer, or output NIL if there is no appropriate candidate.

Formally, we assume that we have access to a snapshot of Wikipedia, in some language en[2], where $en \in X$, X being the set of all languages in Wikipedia, as our knowledge-base KB_{en} with titles also known as links denoted by $L_1, \ldots, L_N$. We can define the goal of Entity Linking (EL) as, given a textual mention m and a document D, $m \in D$ and $m, D \in en$, to identify the best link l_i:

$$\hat{l}^{(m)} = \arg\max_j P(l_j^{(m)}|m, D) \qquad (1)$$

Since computing $P\left(l_j^{(m)}|m, D\right)$ can be prohibitive over

---

[2]Deliberately using the symbol en as it is the most widely chosen language in EL research.

large datasets, we change the problem into computing

$$\hat{l}^m = \arg\max_j P(C|m, D, l_j^{(m)}) \qquad (2)$$

where C is a Boolean variable that measures how "consistent" the pairs (m, D) and $l_j^{(m)}$ are. As a further simplification, given (m, D), we perform an Information Retrieval (IR)-flavored *fast match* to identify the most likely candidate links $l_{j_1}^{(m)}, \ldots, l_{j_m}^{(m)}$ for the input (m, D), then find the $\arg\max$ over this subset.

In cross-lingual EL, we assume that $m, D \in tr$, where tr is some foreign language like Spanish or Chinese. However, we need to link m to some target link $l_i^{(m)}$, where $l_i^{(m)} \in KB_{en}$.

Fast Match Search

The goal of the fast match search is to provide a set of candidates that can be re-scored to compute the $\arg\max$ in Equation (2). To be able to do this, we prepare an anchor-title index, computed from our Wikipedia snapshot, that maps each distinct hyper-link anchor text to its target Wikipedia titles *e.g.* the anchor text "Titanic" is used in Wikipedia to refer both to the famous ship and to the movie. To retrieve the disambiguation candidates l_i for a query mention m, we query the anchor-title index that we constructed. l_i is taken to be the set of titles most frequently linked to with anchor text m in Wikipedia. For cross-lingual EL, in addition to using the English Wikipedia index (built from the English snapshot), we also build an anchor-title index from the respective target language Wikipedia. Once we have that index, we rely on the inter-language links in Wikipedia to map all the non-English titles back to English. Hence, we have an additional anchor-title index where we have foreign hyper-links as surface forms but English titles as the targets e.g. the surface form "Estados Unidos" will have the candidate title *United_States* which is a title in the English Wikipedia.

Embeddings

Before delving into the model architecture, we briefly describe the word embeddings used in this work. Since we are interested in performing cross-lingual EL, we make use of multi-lingual word embeddings, as shown below.

Monolingual Word Embeddings

We use the widely used CBOW word2vec model (Mikolov et al. 2013) to generate English mono-lingual word embeddings.

Multi-lingual Embeddings

Canonical Correlation Analysis (CCA): This technique is based on (Faruqui and Dyer 2014) who learn vectors by first performing SVD on text in different languages, then applying CCA on pairs of vectors for the words that align in parallel corpora. For cross-lingual EL, we use the embeddings provided by (Tsai and Roth 2016), built using the title mapping obtained from inter-language links in Wikipedia.

MultiCCA: Introduced by (Ammar et al. 2016) this technique builds upon CCA and uses a linear operator to project pre-trained monolingual embeddings in each language (except English) to the vector space of pre-trained English word embeddings.

Weighted Least Squares (LS): Introduced by (Mikolov, Le, and Sutskever 2013), the foreign language embeddings are directly projected onto English, with the mapping being constructed through multivariate regression.

Wikipedia Link Embeddings

We are also interested in embedding entire Wikipedia pages (links). In previous work, (Francis-Landau, Durrett, and Klein 2016) run CNNs over the entire article and output one fixed-size vector. However, we argue that this operation is too expensive, and it becomes more expensive for some very long pages (based on our experiments on the validation data). We propose a simpler, less expensive route of modeling the Wikipedia page of a target entity. For every Wikipedia title and using pre-trained word embeddings (obtained in Section), we compute a weighted average of all the words in the Wikipedia page text. We use the inverse document frequency (IDF) of each word as a weight for its vector, to reduce the influence of frequent words. We compute the Wikipedia page embedding for page p (e_p) as:

$$e_p = \frac{\sum_{w \in p} e_w idf_w}{\sum_{w \in p} idf_w} \tag{3}$$

where e_w and idf_w are the embedding vector and the IDF for word w respectively. We further apply (and train) a fully connected *tanh* activation layer to the embedding obtained this way, in order to allow the model to bring the mention context and the Wikipedia link embedding to a similar space before further processing.

Modeling Contexts

In this Section, we will describe how we build the sub-networks that encode the representation of query mention m in the given query document D. This representation is then compared with the page embedding (through cosine similarity) and the result is fed into the higher network (Figure 2).

Noting that the entire document D might not be useful [3] for disambiguating m, we choose to represent the mention m based only on the surrounding sentences of m in D, in contrast to (He et al. 2013; Francis-Landau, Durrett, and Klein 2016), which chose to use the entire document for modeling.

Hence, following similar ideas in (Barrena et al. 2014; Lee et al. 2012), we run a coreference resolution system (Luo et al. 2004) and assume a "one link per entity" paradigm (similar to one sense per document (Gale, Church, and Yarowsky 1992; Yarowsky 1993)). We then use these to build a *sentence-based* context representation of m as well as its finer-grained context encoding, from only words within a window surrounding the mention occurrences.

---

[3] Our experiments on the validation datasets made us think this way.

Modeling Sentences

We collect all the sentences that contain the mention or are part of the entity's coreference chain. Then we combine these sentences together and form a sequence of sentences containing all instances of mention m. We use a convolutional neural network (CNN) to produce fixed-size vector representations from the variable length sentences. We first embed each word into a d-dimensional vector space using the embedding techniques described in the previous section . This results in a sequence of vectors $w_1,...,w_n$. We then map those words into a fixed-size vector using a Convolutional Neural Network (CNN) parameterized with a filter bank $V \in \mathbb{R}^{k \times dc}$, where c is the width of the convolution (unigram, bigram, etc.) and k is the number of filter maps. We apply a *tanh* nonlinearity and aggregate the results with *mean*-pooling. A similar CNN is used for building representations of the first paragraphs of a Wikipedia page which is taken to be the context of the candidate link. First paragraphs of an entity's Wikipedia page consists of one or more sentences. Note that this is different than running CNNs on the whole Wikipedia link embeddings described earlier.

Fine-grained context modeling

While representing the context of a mention as the output of a CNN running over the sentences surrounding it, might allow for relevant patterns to fire, it is not clear if this type of a representation allows for finer-grained meaning distinctions. Furthermore, this does not exploit the fact that words closer to a mention, are stronger indicators of its meaning than words that are far away. Consider, for example, this sentence: *"Ahmadinejad , whose country has been accused of stoking sectarian violence in Iraq, told ABC television that he did not fear an attack from the United States."* If our query mention is ABC, only several words surrounding it are needed for a system to infer that ABC is referring to the *American Broadcasting Company* (a television network), while modeling the entire sentence might lead to losing that signal.

For that purpose, we consider *context* to be the words surrounding a mention within a window of length *n*. For our experiments, we chose *n* to be 4. We collect all the left and right contexts separately, the left ending with the mention string and the right beginning with the mention string.

In a first step, we run LSTMs on these contexts as follows: we run forward LSTMs on the left and backward on the right context and use element-wise mean pooling as the combination strategy. To detail: using the condensed notations of (Cheng, Dong, and Lapata 2016), we run a forward LSTM network over each left context, and a backward LSTM network over each right context, and pool them over all the contexts of each mention. The resulting condensed representations are averaged and then combined using a neural tensor network, using the equation below (also see Figure 1).

$$NTN(l,r;W) = f(\begin{bmatrix} l \\ r \end{bmatrix}^t W^{\{1,...,k\}} \begin{bmatrix} l \\ r \end{bmatrix}) \tag{4}$$

Here l and r are the representations for the overall left and right context ($l, r \in \mathbb{R}^d$), W is a tensor with k slices with
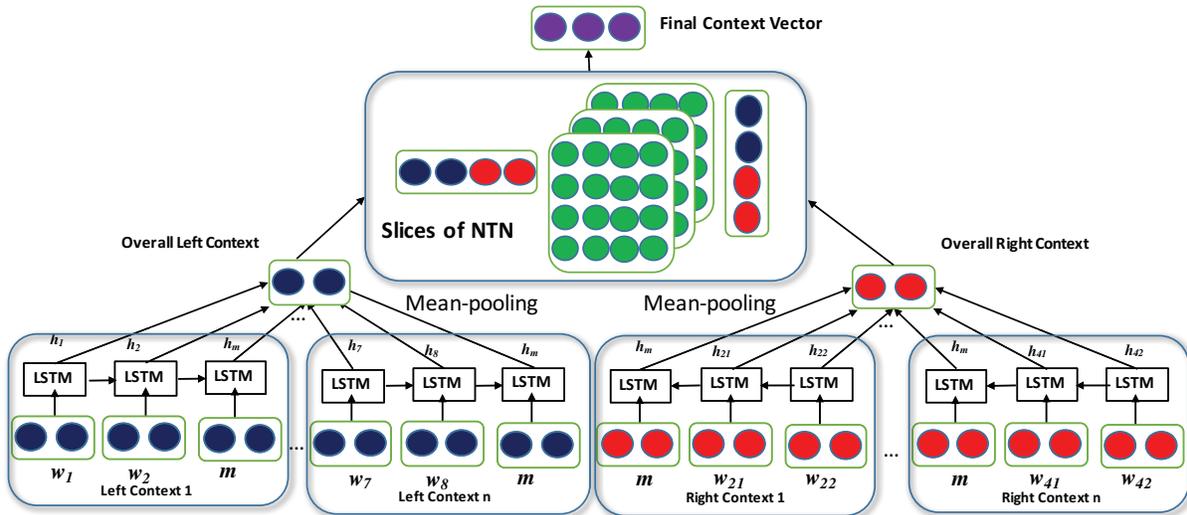
Figure 1: Modeling of fine grained context using LSTMs and NTNs from the left and right contexts obtained from the coreference chain of the query entity.

$W^i \in \mathbb{R}^{2d \times 2d}$, f is a standard nonlinearity applied element wise (sigmoid in our case). The output of NTN is a vector $NTN(l, r; W) \in \mathbb{R}^{k}$ [4].

Cross-Lingual Neural Entity Linking

Neural Model Architecture

The general architecture of our neural EL model is described in Figure 2. Our target is to perform "zero shot learning" (Socher et al. 2013b; Palatucci et al. 2009) for cross-lingual EL. Hence, we want to train a model on English data and use it to decode in any other language, provided we have access to multi-lingual embeddings from English and the target language. We allow the model to compute several similarity/coherence *scores S* (feature abstraction layer): which are several measures of similarity of the context of the mention m in the query document and the context of the candidate link's Wikipedia page, described in details in the next section, which are fed to a feed-forward neural layer H with weights W_h, bias b_h, and a *sigmoid* non-linearity. The output of H (denoted as h) is computed according to $h = \sigma(W_h S + b_h)$. The output of the binary classifier $p(C|m, D, l)$ is the softmax over the output of the final feed-forward layer O with weights W_0 and bias b_0. $p(C|m, D, L)$ represents the probability of the output class C taking a value of 1 (correct link) or 0 (incorrect link), and is computed as a 2 dimensional vector and given by:

$$p(C|m, D, l) = softmax(W_0 h + b_0) \quad (5)$$

Feature Abstraction Layer

In this layer, we encode the similarity between the context of the mention in the source document and the context of

---
[4]We use l to denote left context here for simplicity even when we have used it before to denote a link.
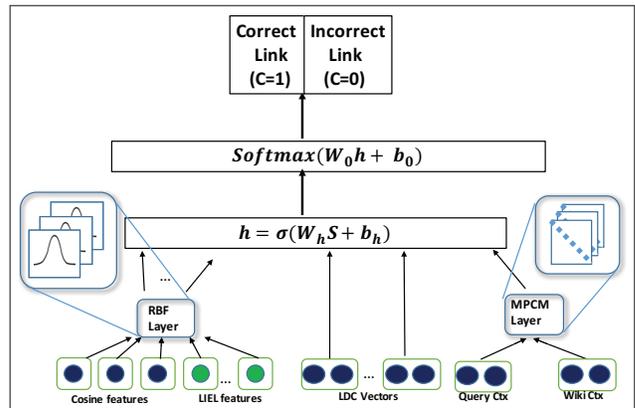

Figure 2: Architecture of our neural EL system. The input to the system are: a document D containing the query mention m and the corresponding Wikipedia candidate link $l_i \in L$, where L is the set of all possible links extracted from the fast match step described in Section .

the corresponding candidate Wikipedia links as obtained through fast match at multiple granularities, described below.

A. Similarity Features by comparing Context Representations

1. "Sentence context - Wiki Link" Similarity: The first input to this layer is the cosine similarity between the CNN representations of its *relevant context* sentences and the embedding of the candidate Wikipedia link (both described in the Embeddings section).

2. "Sentence context - Wiki First Paragraph" Similarity: The next input is the cosine similarity between the CNN

representations of the sentential context of a mention and the first Wikipedia paragraph, following the intuition that often the first paragraph is a concise description of the main content of a page. Multiple sentences are composed using the same model as above.

3. "Fine-grained context - Wiki Link" Similarity: Next, we feed the similarity between the more fine-grained embedding of context described in the Embeddings section, Equation (4) and the embedding of the candidate page link.

4. Within-language Features: We also feed in all the local features described in the LIEL system (Sil and Florian 2016). LIEL uses several features such as "how many words overlap between the mention and Wikipedia title match?" or "how many outlink names of the candidate Wikipedia title appear in the query document?" that compares the similarity of the context of the entity under consideration from the source document and its target Wikipedia page. We also add a feature encoding the probability $P(l_i|m)$, the posterior of a Wikipedia title l_i being the target page for the mention m, using solely the anchor-title index. This feature is a strong indicator to predict if a link l_i is the correct target for mention m.

Multi-perspective Binning Layer: Previous work (Liu et al. 2016) quantizes numeric feature values and then embeds the resulting bins into 10-dimensional vectors. In contrast, we propose a "Multi-perspective Binning Layer" (MPBL) which applies multiple Gaussian radial basis functions to its input, which can be interpreted as a smooth binning process. The above-described similarity values are fed into this MPBL layer, which maps each to a higher dimensional vector. Introducing this layer lets the model learn to respond differently to different values of the cosine input feature, in a neural network friendly way. Our technique differs from (Liu et al. 2016) in that it is able to automatically learn the important regions for the input numeric values.

B. Semantic Similarities and Dissimilarities

1. Lexical Decomposition and Composition (LDC): We use the recently proposed LDC[5] model in (Wang, Mi, and Ittycheriah 2016) to compare the contexts. For brevity, we only give a brief description of this feature - we direct the reader to the original paper. We represent the source context S and the Wikipedia paragraph T as a sequence of pre-trained embeddings of words. S = $[s_1, \ldots, s_m]$ and T=$[t_1, \ldots, t_n]$ where s_i and t_j are the pre-trained word embeddings for the ith and jth word from the source context and the Wikipedia paragraph respectively. The steps of LDC are summarized below. For each word s_i in S, the semantic matching step finds a matching word $\hat{s}_i$ from T. In the reverse direction, a matching word $\hat{t}_j$ is found for each t_j in T. For a word embedding, its matching word is the one with the highest cosine similarity. Hence, $\hat{s}_i = t_k$ where k = $\arg\max_j \cos(s_i, t_j)$ and $\hat{t}_j = s_k$ where k = $\arg\max_i \cos(t_j, s_i)$.

$$\hat{s}_i = t_k \text{ where k} = \arg\max_j \cos(s_i, t_j)$$

$$\hat{t}_j = s_k \text{ where k} = \arg\max_i \cos(t_j, s_i)$$

The next step is decomposition, where each word embedding s_i (or t_j) is decomposed based on its semantic matching vector $\hat{s}_i$ (or $\hat{t}_j$) into two components: similar component s_i^+ (or t_j^+) and dissimilar component s_i^- (or t_j^-). We compute the cosine similarity between s_i and $\hat{s}_i$ (or t_i and $\hat{t}_i$) and decompose linearly. Hence, $(s_i^+, s_i^-) = (\alpha s_i, (\sqrt{1 - \alpha^2})s_i)$ and $(t_i^+, t_i^-) = (\alpha t_i, \sqrt{1 - \alpha^2}t_i)$ where $\alpha = \cos(s_i, \hat{s}_i)$ and $\alpha = \cos(t_i, \hat{t}_i)$.

$$(s_i^+, s_i^-) = (\alpha s_i, (1 - \alpha)s_i) \text{ and } \alpha = \cos(s_i, \hat{s}_i)$$

$$(t_i^+, t_i^-) = (\alpha t_i, (1 - \alpha)t_i) \text{ and } \alpha = \cos(t_i, \hat{t}_i)$$

In the Composition step, the similar and dissimilar components are composed at different granularities using a two channel CNN and pooled using *max*-pooling. The output vector is the representation of the similarity (and dis-similarity) of the source context of the mention with the Wikipedia page of the target entity.

2. Multi-perspective Context Matching (MPCM): Next, we input a series of weighted cosine similarities between the query mention context and the Wikipedia link embedding, as described in (Wang et al. 2016). Our argument is that while cosine similarity finds semantically similar words, it has no relation to the classification task at hand. Hence, we propose to train weight vectors to re-weigh the dimensions of the input vectors and then compute the cosine similarity. The weight vectors will be trained to maximize the performance on the entity linking task. We run CNNs to produce a fixed size representations for both query and candidate contexts from Section . We build a node computing the cosine similarity of these two vectors, parametrized by a weight matrix. Each row in the weight matrix is used to compute a score as $u_k = \cos(w_k \circ v_1, w_k \circ v_2)$, where v_1 and v_2 are input d dimensional vectors, $w_k \in \mathbb{R}^d$ is the kth column in the matrix, **u** is a l-dimensional output vector, and $\circ$ denotes a element-wise multiplication. Note that re-weighting the input vectors is equivalent to applying a diagonal tensor with non-negative diagonal entries to the input vectors.

Training and Decoding

To train the model described in Equation (2), the binary classification training set is prepared as follows. For each mention $m_{ij} \in D_i$ and its corresponding correct Wikipedia page $l^{(m_{ij})}$, we use our fast match strategy (discussed in Page 2) to generate K_{ij} number of incorrect Wikipedia pages $(l_{ij_k})_k$. l_i and l_{ij_k} represent positive and negative examples for the binary classifier. Pairs in the list of $[(m_{ij}, D, l_{ij}), (m_{ij}, D, l_{ij_0}), \ldots, (m_{ij}, D_i, l_{ij_{K_{ij}}})]$ will be used to produce the similarity/ dis-similarity vectors S_{ij_k}. Classification label Y_{ij_k} that corresponds to input vector (m_{ij}, D_i, l_{ij_k}) will take the value of 1 for the correct Wikipedia page and 0 for incorrect ones. The binary classifier is trained with the training set T which contains all the

(m, D, l, Y) data pairs[6].

Training is performed using stochastic gradient descent on the following loss function:

$$-\frac{1}{|T|} \sum_{(m_j, D_j, l_j, Y_j) \in T} \log P(C = Y_j | m_j, D_j, l_j) \quad (6)$$

Decoding a particular mention $m \in D$, is simply done by running fast match to produce a set of likely candidate Wikipedia pages, then generate the system output $l^{(\hat{m})}$ as in Equation (2).

Note that the model does all this by only computing similarities between texts in the same language, or by using cross-lingual embeddings, allowing it to transcend across languages.

Experiments

We evaluate our proposed method on the benchmark datasets for English: CoNLL 2003 and TAC 2010 and Cross-Lingual: TAC 2015 Trilingual Entity Linking dataset.

Datasets

English (CoNLL & TAC): The CoNLL dataset (Hoffart et al. 2011) contains 1393 articles with about 34K mentions, and the standard performance metric is mention-averaged accuracy. The documents are partitioned into train, test-a and test-b. Following previous work, we report performance on the 231 test-b documents with 4483 linkable mentions. The TAC 2010 source collection includes news from various agencies and web log data. Training data includes a specially prepared set of 1,500 web queries. Test data includes 2,250 queries – 1,500 news and 750 web log uniformly distributed across person, organisation, and geo-political entities.

Cross-Lingual (TAC): We evaluate our method on the TAC 2015 Tri-Lingual Entity Linking datasets which comprises of 166 Chinese documents (84 news and 82 discussion forum articles) and 167 Spanish documents (84 news and 83 discussion forum articles). The mentions in this dataset are all named entities of five types: Person, Geo-political Entity, Organization, Location, and Facility.

We use standard train, validation and test splits if the datasets come with it, else we use the CoNLL validation data as dev. For the CoNLL experiments, in addition to the Wikipedia anchor-title index, we also use a alias-entity mapping previously used by (Pershina, He, and Grishman 2015; Globerson et al. 2016; Yamada et al. 2016). We also use the mappings provided by (Hoffart et al. 2011) obtained by extending the "means" tables of YAGO (Hoffart et al. 2013).

Hyperparameters

We tune all our hyper-parameters on the development data. We run CNNs on the sentences and the Wikipedia embeddings with filter size of 300 and width 2. The non-linearity used is *tanh*. For both forward (left) and backward (right) LSTMs, we use mean pooling. We tried max-pooling and also choosing the last hidden state of the LSTMs but mean

---

[6] The ratio of positive to negative training events is controlled to produce a more balanced training data.

---

pooling worked the best. We combine the LSTM vectors for all the left and all the right using mean pooling, as well. For the NTNs, we use sigmoid as the non-linearity and an output size of 10 and use $L2$ regularization with a value of 0.01. Finally, to compute the similarity we feed the output of the NTN to another hidden layer with sigmoid non-linearity for a final output vector of size 300. For the main model, we again use sigmoid non-linearity and an output size of 1000 with a dropout rate of 0.4. We do not update the Wikipedia page embeddings as they did not seem to provide gains in numbers while testing on development data. We also do not update the multi-lingual embeddings for the cross-lingual experiments. For the English experiments, we update the mono-lingual English word embeddings. For the MPBL node, the number of dimensions is 100.

Comparison with the SOTA

The current SOTA for English EL are (Globerson et al. 2016) and (Yamada et al. 2016). We also compare with LIEL (Sil and Florian 2016) which is a language-independent EL system and has been a top performer in the TAC annual evaluations. For cross-lingual EL, our major competitor is (Tsai and Roth 2016) who uses multi-lingual embeddings similar to us. We also compare with several other systems as shown in Table 1a, 1b and 2a along with the respective top ranked TAC systems.

English Results

Table 1a shows our performance on the CoNLL dataset along with recent competitive systems in terms of micro-average accuracy. We outperform (Globerson et al. 2016) by an absolute average of 1.27% and (Yamada et al. 2016) by 0.87%. Globerson et al. use a multi-focal attention model to select specific context words that are essential for linking a mention. Our model with the lexical decomposition and composition and the multi-perspective context matching layers seems to be more beneficial for the task of EL.

Table 1b shows our results when compared with the top systems in the evaluation along with other SOTA systems on the TAC2010 dataset. Encouragingly, our model's performance is slightly better than the top performer, Globerson (2016), and outperforms both the top rankers from this challenging annual evaluation by 8% absolute percentage points. Note that in both the datasets, our model obtains 7.77% (on CoNLL) and 8.75% (on TAC) points better than (Sil and Florian 2016), which is a SOTA multi-lingual system. Another interesting fact we observe is that our full model outperforms (Sun et al. 2015) by 3.5% points, where they employ NTNs to model the semantic interactions between the context and the mention. Our model uses NTNs to model the left and right contexts from the full entity coreference chain in a novel fashion not used previously in the EL research and seems highly useful for the task. Interestingly, we observe that the recent (Gupta, Singh, and Roth 2017) EL system performs rather poorly on the CoNLL dataset (7.5% lower than our model) even when their system employ entity type information from a KB which our system does not.

While doing ablation study, we notice that adding the LDC layer provides a boost to our model in both the datasets,

Systems	In-KB acc. %
Hoffart et al. (2011)	82.5
Gupta et al. (2017)	82.9
He et al. (2013)	85.6
Francis-Landau et al. (2016)	85.5
Sil & Florian (2016)	86.2
Lazic et al. (2015)	86.4
Chisholm & Hachey (2015)	88.7
Ganea et al. (2015)	87.6
Pershina et al. (2015)	91.8
Globerson et al. (2016)	92.7
Yamada et al. (2016)	93.1
This work	92.1
This work+CtxLSTMs	93.0
This work+CtxLSTMs+LDC	93.4
This work+CtxLSTMs+LDC+MPCM	**94.0**

(a) CoNLL2003

Systems	In-KB acc. %
TAC Rank 1	79.2
TAC Rank 2	71.6
Sil & Florian (2016)	78.6
He et al. (2013)	81.0
Chisholm & Hachey (2015)	80.7
Sun et al. (2015)	83.9
Yamada et al. (2016)	85.2
Globerson et al. (2016)	87.2
This work	85.0
This work+CtxLSTMs	86.3
This work+CtxLSTMs+LDC	86.9
This work+CtxLSTMs+LDC+MPCM	**87.4**

(b) TAC2010

Table 1: Performance comparison on the CoNLL 2003 testb and TAC2010 datasets. Our system outperforms all EL systems, including the only other multi-lingual system, (Sil and Florian 2016).

and the multi-perspective context matching (MPCM) layer provides an additional 0.5% (average) points improvement. We see that adding in the context LSTM based layer (fine-grained context) adds almost 1% point (in both the datasets) over the base similarity features.

Cross-lingual Results

Spanish: Table 2a shows our performance on cross-lingual EL on the TAC2015 Spanish dataset. The experimental setup is similar as in the TAC diagnostic evaluation, where systems need to predict a link as well as produce the type for a query mention. We use an entity type classifier to attach the entity types to the predicted links as described in our previous work in (Sil, Dinu, and Florian 2015). We compare our performance to (Sil and Florian 2016), which was the top ranked system in TAC 2015, and the cross-lingual wikifier (Tsai and Roth 2016). We see that our zero-shot model trained with the multi-CCA embeddings is 1.32% and 1.85% percentage points better than the two competitors respectively.
Chinese: Table 2b displays our performance on the TAC2015 Chinese dataset. Our proposed model is 0.73% points better than (Tsai and Roth 2016). In both cross-lingual experiments, the multi-CCA embeddings outperform LS and CCA methods. In Spanish, LS and CCA are tied but in Chinese, CCA performs better than LS. Note that "this work" in Table 2 indicates our full model with LDC and MPCM.

Related Work

Previous works in EL (Bunescu and Pasca 2006; Mihalcea and Csomai 2007) involved finding the similarity of the context in the source document and the context of the candidate Wikipedia titles. Recent research on EL has focused on sophisticated global disambiguation algorithms (Globerson et al. 2016; Milne and Witten 2008; Cheng and Roth 2013;

Sil and Yates 2013) but are more expensive since they capture coherence among titles in the given document. However, (Ratinov et al. 2011) argue that global systems provide a minor improvement over local systems. Our proposed EL system is a local system which comprises of a deep neural network architecture with various layers computing the semantic similarity of the source documents and the potential entity link candidates modeled using techniques like neural tensor network, multi-perspective cosine similarity and lexical composition and decomposition.

Sun et al. (2015) used neural tensor networks for entity linking, between mention and the surrounding context. But this did not give good results in our case. Instead, the best results were obtained by composing the left and right contexts of all the mentions in the coreference chain of the target mention. In this work, we also introduced state-of-the-art similarity models like MPCM and LDC for entity linking. Combination of all these components helps our model score 3.5 absolute accuracy improvement over Sun et al. (2015).

The cross-lingual evaluation at TAC KBP EL Track that started in 2011 (Ji, Grishman, and Dang 2011; Ji et al. 2015) has Spanish and Chinese as the target foreign languages. One of the top performers (Sil and Florian 2016), like most other participants, perform EL in the foreign language (with the corresponding foreign KB), and then find the corresponding English titles using Wikipedia inter-language links. Others (McNamee et al. 2011) translate the query documents to English and do English EL. The first approach relies on a large enough KB in the foreign language, whereas the second depends on a good machine translation system. Similar to (Tsai and Roth 2016), the ideas proposed in this paper make significantly simpler assumptions on the availability of such resources, and therefore can also scale to lower resource languages, while doing very well also on high-resource languages. However, unlike our model they need to train and decode the model on the target language. Our model once trained on English can perform

Systems	Linking Acc %
Sil & Florian (2016) / TAC Rank 1	80.4
Tsai & Roth (2016)	80.9
This Work (LS)	81.9
This Work (CCA)	81.8
This Work (Multi-CCA)	**82.3**

(a) Spanish

Systems	Linking Acc %
TAC Rank 1	83.1
Tsai & Roth (2016)	83.6
This Work (LS)	84.1
This Work (CCA)	84.3
This Work (Multi-CCA)	**84.4**

(b) Chinese

Table 2: Performance comparison on the TAC 2015 Spanish and Chinese datasets. Our system outperforms all the previous EL systems.

cross-lingual EL on any target language.

Some recent work involves (Lin, Lin, and Ji 2017) but is unrelated since it solves a different problem (EL from only lists) than generic EL and hence an apples-apples comparison cannot be done. (Pan et al. 2017) is related but their method prefers common popular entities in Wikipedia and they select training data based on the topic of the test set. Our proposed method is more generic and robust as it is once trained on the English Wikipedia and tested on any other language without re-training. (Tan et al. 2017) solves a different problem by performing EL for queries while we perform EL for generic documents like news. Recently (Gupta, Singh, and Roth 2017) propose an EL system by jointly encoding types from a knowledge-base. However, their technique is limited to only English and unlike us do not perform cross-lingual EL.

Conclusion

Recent EL research, that we compare against, have produced models that achieve either SOTA mono-lingual performance or cross-lingual performance, but not both. We produce a model that performs zero-shot learning for the task of cross-lingual EL: once trained on English, the model can be applied to any language, as long as we have multi-lingual embeddings for the target language. Our model makes effective use of the similarity models (LDC, MPCM) and composition methods (neural tensor network) to capture similarity/dissimilarity between the query mention's context and the target Wikipedia link's context. We test three methods of generating multi-lingual word embeddings and determine that the MultiCCA-generated embeddings perform best for the task of EL for both Spanish and Chinese. Our model has strong experimental results, outperforming all the previous SOTA systems in both mono and cross-lingual experiments. Also, with the increased focus on cross-lingual EL in future TAC evaluations, we believe that this zero-shot learning technique would prove useful for low-resource languages: train one model and use it for any other language.

Acknowledgements

References

Ammar, W.; Mulcaire, G.; Tsvetkov, Y.; Lample, G.; Dyer, C.; and Smith, N. A. 2016. Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925*.

Barrena, A.; Agirre, E.; Cabaleiro, B.; Penas, A.; and Soroa, A. 2014. "one entity per discourse" and "one entity per collocation" improve named-entity disambiguation. In *COLING*, 2260–2269.

Bunescu, R., and Pasca, M. 2006. Using encyclopedic knowledge for named entity disambiguation. In *EACL*.

Cheng, X., and Roth, D. 2013. Relational inference for wikification. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Cheng, J.; Dong, L.; and Lapata, M. 2016. Long short-term memory-networks for machine reading. *CoRR* abs/1601.06733.

Cucerzan, S. 2007. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, 708–716.

Faruqui, M., and Dyer, C. 2014. Improving vector space word representations using multilingual correlation. ACL.

Francis-Landau, M.; Durrett, G.; and Klein, D. 2016. Capturing semantic similarity for entity linking with convolutional neural networks. In *Proc. NAACL 2016*.

Gale, W. A.; Church, K. W.; and Yarowsky, D. 1992. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, 233–237.

Globerson, A.; Lazic, N.; Chakrabarti, S.; Subramanya, A.; Ringgaard, M.; and Pereira, F. 2016. Collective entity resolution with multi-focal attention. *ACL*.

Gupta, N.; Singh, S.; and Roth, D. 2017. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2671–2680.

He, Z.; Liu, S.; Li, M.; Zhou, M.; Zhang, L.; and Wang, H. 2013. Learning entity representation for entity disambiguation. In *ACL (2)*, 30–34.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hoffart, J.; Yosef, M. A.; Bordino, I.; Furstenau, H.; Pinkal, M.; Spaniol, M.; Taneva, B.; Thater, S.; and Weikum1, G. 2011. Robust Disambiguation of Named Entities in Text. In *EMNLP*, 782–792.

Hoffart, J.; Suchanek, F. M.; Berberich, K.; and Weikum, G. 2013. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence* 194:28–61.

Ji, H.; Dang, H.; Nothman, J.; and Hachey, B. 2014. Overview of tac-kbp2014 entity discovery and linking tasks. In *Proc. Text Analysis Conference (TAC2014)*.

Ji, H.; Nothman, J.; Hachey, B.; and Florian, R. 2015. Overview of tac-kbp2015 tri-lingual entity discovery and linking. In *TAC*.

Ji, H.; Grishman, R.; and Dang, H. T. 2011. Overview of the tac2011 knowledge base population track. *TAC*.

Lee, H.; Recasens, M.; Chang, A.; Surdeanu, M.; and Jurafsky, D. 2012. Joint entity and event coreference resolution across documents. In *EMNLP*, 489–500.

Lin, Y.; Lin, C.-Y.; and Ji, H. 2017. List-only entity linking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, 536–541.

Liu, D.; Lin, W.; Zhang, S.; Wei, S.; and Jiang, H. 2016. Neural networks models for entity discovery and linking. *arXiv preprint arXiv:1611.03558*.

Luo, X.; Ittycheriah, A.; Jing, H.; Kambhatla, N.; and Roukos, S. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *ACL*, 135.

McNamee, P.; Mayfield, J.; Oard, D. W.; Xu, T.; Wu, K.; Stoyanov, V.; and Doermann, D. 2011. Cross-language entity linking in maryland during a hurricane.

Mihalcea, R., and Csomai, A. 2007. Wikify!: Linking documents to encyclopedic knowledge. In *CIKM*, 233–242.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T.; Le, Q. V.; and Sutskever, I. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Milne, D., and Witten, I. H. 2008. Learning to link with wikipedia. In *CIKM*.

Palatucci, M.; Pomerleau, D.; Hinton, G. E.; and Mitchell, T. M. 2009. Zero-shot learning with semantic output codes. In *Advances in neural information processing systems*, 1410–1418.

Pan, X.; Zhang, B.; May, J.; Nothman, J.; Knight, K.; and Ji, H. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 1946–1958.

Pershina, M.; He, Y.; and Grishman, R. 2015. Personalized page rank for named entity disambiguation. In *HLT-NAACL*, 238–243.

Ratinov, L.; Roth, D.; Downey, D.; and Anderson, M. 2011. Local and global algorithms for disambiguation to wikipedia. In *ACL*.

Sil, A., and Florian, R. 2016. One for all: Towards language independent named entity linking. *ACL*.

Sil, A., and Yates, A. 2013. Re-ranking for Joint Named-Entity Recognition and Linking. In *CIKM*.

Sil, A.; Dinu, G.; and Florian, R. 2015. The ibm systems for trilingual entity discovery and linking at tac 2015. In *Proceedings of the Eighth Text Analysis Conference (TAC2015)*.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. 2013a. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, 926–934.

Socher, R.; Ganjoo, M.; Manning, C. D.; and Ng, A. 2013b. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, 935–943.

Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013c. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, volume 1631, 1642.

Sun, Y.; Lin, L.; Tang, D.; Yang, N.; Ji, Z.; and Wang, X. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. In *IJCAI*, 1333–1339.

Tan, C.; Wei, F.; Ren, P.; Lv, W.; and Zhou, M. 2017. Entity linking for queries by searching wikipedia sentences. *arXiv preprint arXiv:1704.02788*.

Tsai, C.-T., and Roth, D. 2016. Cross-lingual wikification using multilingual embeddings. In *Proceedings of NAACL-HLT*, 589–598.

Wang, Z.; Mi, H.; Hamza, W.; and Florian, R. 2016. Multi-perspective context matching for machine comprehension. *arXiv preprint arXiv:1612.04211*.

Wang, Z.; Mi, H.; and Ittycheriah, A. 2016. Sentence similarity learning by lexical decomposition and composition. *COLING*.

Yamada, I.; Shindo, H.; Takeda, H.; and Takefuji, Y. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. *arXiv:1601.01343*.

Yarowsky, D. 1993. One sense per collocation. In *Proceedings of the workshop on Human Language Technology*, 266–271.