Cross Lingual Mention and Entity Embeddings for Cross-Lingual Entity Disambiguation

Hamed Shahbazi, Chao Ma, Xiaoli Fern and Prasad Tadepalli
School of Electrical Engineering and Computer Science, Oregon State University
{ shahbazh, machao, xfern, tadepall }@eecs.oregonstate.edu

Abstract

Cross-lingual Entity Discovery and Linking (EDL) task involves discovering query mentions in cross-lingual documents and linking them to their referent entities in an English Knowledge Base (KB). Traditional entity Linking models heavily rely on engineering manual and often language dependent features. Recently, deep learning based models have emerged as compelling solutions that alleviate the problem of feature engineering. In this paper we propose a deep learning based model in which the cross-lingual contextual information are encoded into mention and entity embeddings and then hard-wired into a linker that can optionally leverages them along with other lexical features in its disambiguation process. The experimental results show that the embeddings can efficiently enhance the performance of the linker while they are learned without relying on specific language features.

1 Introduction

In the cross-lingual Entity Discovery and Linking (EDL) task, cross-lingual mentions are identified and then linked to either referent entities in the Knowledge Base (KB) or NIL. Recent deep learning based models [Lee *et al.*, 2011], [He *et al.*, 2013], and [Tsaia and Roth, 2016] have alleviated the problem of feature engineering in the traditional models by encoding contextual information existing in the structured or unstructured data into embeddings.

In this paper we propose a cross-lingual entity linking model in which we use deep learning techniques to make the performance less sensitive to language specifics. Our proposed cross-lingual entity linker consists of mention and context models. The mention model captures the lexical compatibility between mention and entity in the English domain. On the other hand the context model leverages the contextual information encoded in mention and entity embeddings to make mention model less sensitive to English-dependent features.

Our mention model uses transliteration to obtain the mention-entity features when the mention is in non-English language. In order to improve the performance of the mention model, similar to [Durrett and Klein, 2015] we define a latent query variable for each mention. The domain of each latent query variable is the most probable prefixes of the mention up to the head with optional truecasing and lemmatization. The weights of the lexical features are adapted to the TAC 2015 training set.

In our context model, mention and entity embeddings are learned by applying Skip-Gram model [Mikolov *et al.*, 2013] on a corpus comprised from English, Chinese and Spanish Wikipedia textual dumps.

The preliminary evaluations of our submissions to the TAC 2016 are reported in this paper. The experiments results show that the context model is effective in improving the performance without imposing language specific constraints.

2 Entity Discovery

We use our last year developed annotator proposed at TAC KBP 2015 to annotate the documents. This module runs a pipeline of sentence splitting, tokenizing (chunking), POS and NER tagging to finally generate named entity spans and types for the given document in English, Chinese and Spanish languages.

Our annotator uses pre-trained models from Stanford CoreNLP [Manning *et al.*, 2014] and OpenNLP imported in the Reconcile system [Stoyanov *et al.*, 2010]. We make some adjustments to the module to meet the new criteria in the TAC KBP 2016 entity discovery such as discovering non-embedded mentions and entity types of PER, ORG, LOC, FAC, and GPE. Additionally we enhance the lexicon feature of the Stanford NER for English to improve the mention extraction recall and the NER types as well.

3 Entity Linking

Given a query document which is a list of words and mentions as $D = \{w_0, ...m_i, ...m_j, ...w_k...w_{T-1}\}$ where w_k is a regular word or phrase and m_i is a query mention; the objective of the linker is to find the optimal assignment of entities to maximize:

$$P(e_1, ...e_n | m_1, ..m_n, D) \quad (1)$$

Where e_i is the assigned entity to mention m_i and n is the number of mentions in the document.

There are multiple ways of addressing the optimization in Eq 1. In the sake of tractability, similar to the simplifying

assumption of the star-model proposed in [Globerson *et al.*, 2016], the inference in our model is decomposed per mention. Moreover as shown in the graphical model in Fig 3 the inference about m_i can also take into account the contextual information from other mentions in the document. This contextual influence from other mentions defines our context model.
Similar to [Durrett and Klein, 2015] for each mention m_i we define a latent query variable q_i. The domain of each variable q_i is different prefixes of the mention m_i containing the head with optional operations like truecasing or lemmatization. The unary and binary factors in this model are defined on query and query-entity variables respectively.
In the following section we describe the inference and learning in our model.

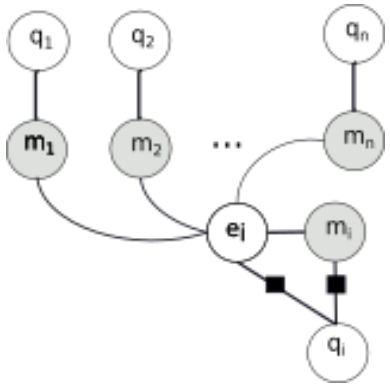

Figure 1: Inference for variable e_i is influenced by other observed mention variables.

3.1 Inference

The score of the candidate entity y_i to be the referent entity for mention m_i is defined as the multiplication of two normalized (soft-max normalization on candidate set of m_i) lexical and contextual factors as follows:

$$s(m_i, y_i; D) = s_l(t(m_i), y_i)\ s_c(m_i, y_i; D) \tag{2}$$

Where $t(m_i)$ is the transliteration of m_i if m_i is in non-English language else m_i. The lexical factor $s_l(t(m_i), y_i)$ is computed based on the lexical compatibility between $t(m_i)$ and entity y_i. The lexical score is in fact the following log-linear probability marginalized on the query variable:

$$s_l(t(m_i), y_i) \sim \sum_q exp(w^T\ f(t(m_i), y_i, q)) \tag{3}$$

Where w and f are lexical weights and features respectively. The contextual compatibility i.e. $s_c(m_i, y_i; D)$ is computed by using the hard-wired cross-lingual embeddings as follows:

$$s_c(m_i, y_i; D) = \sum_{j=1}^{n} \alpha_{i,j} v_j.w_i \tag{4}$$

where v_j and w_i are embeddings for mention m_j and candidate y_i and operation dot (.) is the cosine distance. The scalar $\alpha_{i,j}$ reflects how much mention m_j can make decision for mention m_i and is computed as follows:

$$\alpha_{i,j} = \frac{v_i.v_j}{\sum_{k=1}^{n} v_i.v_k} \tag{5}$$

where v_i and v_j are embeddings for mentions m_i and m_j respectively and n is the number of mentions in the document. In practice we only consider top K closest mentions to m_i. Therefore if $\bar{\alpha_i}$ is the sorted vector of α_i then Eq 4 is updated to:

$$s_c(m_i, y_i; D) = \sum_{j=1}^{K} \bar{\alpha}_{i,j} v_j.w_i \qquad K \leq n \tag{6}$$

The intuition for the above definition is to have only K nearest neighbors of mentions m_i to influence the inference about m_i. In our experiments we use K = 6.

3.2 Learning

We first learn the cross-lingual embeddings for mentions and entities and then having hard-wired them into the Eq 2, we adapt the weight vector of the lexical features to TAC 2015 training set.
In order to learn the cross-lingual embeddings, we create a combined corpus by shuffling and modifying sentences from English, Chinese and Spanish Wikipedia textual dumps. To create the corpus, each sentence S in a dump which is the list of words and pairs of mention-entities (hyperlinks) denoted as [w_0, w_1, ...$m_i|y_i$, ... w_n] is extended to [w_0, $d(w_0)$, w_1, $d(w_1)$, ...m_i, $t(m_i)$, y_i, $p(y_i)$, ... w_n, $d(w_n)$] and added to the combined corpus. Functions d(.), t(.) and p(.) are look-up functions that return the parallel word, transliterated mention and parallel entity in other languages different from the language of S, respectively.
The leaner is a Skip-gram based model; In the learning process if the current word is mention m_i or entity y_i then the negative samples are generated from the candidate set of the mention m_i.
Having learned the embeddings, the weight vector of the lexical features are adapted to the TAC 2015 training set. We use AdaGrad [Duchi *et al.*, 2011] as the optimizer.

4 Candidate Generation

We apply a heuristic method similar to [Durrett and Klein, 2015] to encode the amount of mention-entity compatibility information in raw Wikipedia and Freebase text into an indexing structure. Each pair of mention-hyperlinks like [federal republic | Federal_republic] enhances the heuristic score for the query-entity [America → United States] in the indexer.
In order to improve the candidate identification recall, we exploit query expansion to take care of the abbreviated named mentions and also incomplete and mis-spelled mentions in the query document. The query expansion for only English domain utilizes the information from the corefenence resolver that we apply in the pre-processing stage.

5 NIL Clustering

We develop NIL clusters by applying co-reference resolution on mentions. We firstly form coref clusters within each doc-

ument independently and then merge them across documents as described in sections 5.1 and 5.2.

5.1 Within Document Coreference Resolution

We start by applying the Prune-and-Score algorithm [Ma *et al.*, 2014] for within document coreference resolution. The mentions are processed in a left-to-right order. At each step, a mention m_i either merges with an existing cluster C_k or starts a new cluster C_i (containing only m_i). Parameterized pruning and scoring functions, F_{prune} and F_{score} conditioned on F_{prune} are used to guide the search. The pruning function F_{prune} prunes the list of all possible actions to a size b and the scoring function F_{score} picks the best action from the b actions.
We employ LambdaMART as the ranking model for both functions, and the parameter b is tuned on the development set. The feature definitions are tweaked to handle named mentions. All features specific to pronouns are removed and additional features are defined on the Wiki page of the mentions. The within doc prune-and-score coref system was trained on ACE2004 and ACE2005 datasets.

5.2 Cross Document Coreference Resolution

We implement a rule-based agglomerative clustering algorithm for cross document coreference resolution. Clusters formed by the within-document coreference system are taken as input and rules are applied on each pair of clusters spanning two different documents. Similar to the rule based Stanford multi-sieves system [Lee *et al.*, 2011], our system considers pairs of clusters to perform a merge operation.
While the Stanfords system applies rules in multiple sieves, we apply all the rules together on each pair of clusters. Given a pair of clusters (C_s, C_t), we enumerate all pairs of mentions (m_i , m_j) , where $m_i \in C_s$ and $m_j \in C_t$, and run a rule based scoring function f on each mention pair. The function f returns a boolean value (either 0 or 1), representing the similarity between these two mentions. Finally, we sum over all the pair scores, and divide this sum by the number of pairs as score of this cluster pair. This score represents the proportion of similar mention pairs between two clusters.
We set a threshold h, and merge the pair of clusters if its score is higher than h. The rules applied in function f are as follows:

- ExactStringMatch: True if spans of the two mentions match.
- IsDemonym: True if one mention is the demonym of the other mention, e.g. British and Britain.
- IsAcronym: True if one mention is the acronym of the other mention, e.g. LA and Los Angeles.
- WikiTitleMatch: True if the wiki pages (given by the wikifier) of the two mentions are the same.

If one of the rules above is satisfied, f returns 1 and 0 otherwise. Currently, f can only return binary values. The threshold h is an empirical value that is hand tuned on the development data.

6 Experiments

We conduct experiments on TAC EDL 2016 test split. The test split contains 168, 168 and 167 documents in English, Spanish and Chinese respectively. For the first window of TAC 2016 we have 4 submissions with the following configurations:

- **OSU-DEFT1:** In this run we only use the mention model.
- **OSU-DEFT2:** We enhance the performance of the mention model by adding the contextual information between mention m_i and entity e_i into OSU-DEFT1. Thus in this run the embeddings are hard-wired into Eq 2 and we relearn the mention model.
- **OSU-DEFT3:** This run is the same as OSU-DEFT2 in which we also consider a heuristic score (the score extracted from candidate generation) as a lexical feature.
- **OSU-DEFT4:** Here we set the annotator in the OSU-DEFT2 to generate more nominal mentions. The new nominal mentions are extracted using a pretty huge lexicons.

The preliminary results for the above configurations are shown in Table 1. In the first window of the TAC 2016 we didn't use the full functionality of the embeddings; the contextual dependency was only limited to the pair of mention-entity and other mentions were not included. Moreover an accidental unset flag in the pre-processing caused a low recall for the English mention extraction.
In the second window we fixed the issues in the first window. The last row in Table 1 shows the performance of OSU-DEFT2 but in the second window. Comparing with the OSU-DEFT2 in the first window (RUN 2, WIN 1) we notice significant improvement.
The effect of the contextual information on the precision of the linker in the English domain is shown in Table 2. In this table Model-1 only utilizes the mention model. In Model-2 and Model-3 we exploit the context model without using the mention model. Model-2 is the same as Model-3 except for the value of parameter k where is 1 in Model-2 and 6 in Model-3. Model-2 is in fact similar to the setting in [Blanco *et al.*, 2015] where we only use the cosine similarity between mention and entity as the disambiguation score.
As shown in Table 2 the context model is as capable as the mention model while it doesn't rely on language-dependent features. Moreover when we use $k > 1$ in Model-3 we factor in more evidence in the decision making process leading to higher precision that even beats the mention model.

The candidate generation affects the performance of the linker with two factors; candidate size and ambiguity among candidates. Table 3 shows the probability of the gold position in the candidate set using the heuristic score of the indexer for English domain in TAC 2016 test split. As we can see, the cumulative distribution of gold position for the ranges between (1-5) and (6-10) are about 87% and 1.3% respectively. On the other hands the indexer loses about 10% when it fails to retrieve the true entity from the indexing structure.
One example of the out-of-set mention-entity is pair (Iron Lady, Ellen Johnson Sirleaf). This mention-entity relation is

Measure	WIN	RUN	P	R	F
strong typed link match	1	1	0.510	0.381	0.436
	1	2	0.509	0.372	0.430
	1	3	0.510	0.382	0.437
	1	4	0.507	0.372	0.429
	2	2	0.569	0.460	0.508
strong link match	1	2	0.556	0.415	0.475
	1	2	0.552	0.404	0.466
	1	3	0.556	0.416	0.476
	1	4	0.551	0.404	0.466
	2	2	0.594	0.480	0.530
strong typed nil match	1	1	0.045	0.013	0.021
	1	2	0.045	0.013	0.021
	1	3	0.045	0.013	0.021
	1	4	0.064	0.043	0.051
	2	2	0.045	0.013	0.021
strong typed all match	1	1	0.462	0.299	0.363
	1	2	0.117	0.046	0.066
	1	3	0.462	0.300	0.364
	1	4	0.416	0.298	0.348
	2	2	0.514	0.367	0.429
strong linked mention match	1	1	0.660	0.494	0.565
	1	2	0.666	0.487	0.563
	1	3	0.660	0.494	0.5655
	1	4	0.665	0.487	0.563
	2	2	0.706	0.571	0.631
strong typed mention match	1	1	0.552	0.358	0.434
	1	2	0.552	0.358	0.434
	1	3	0.552	0.358	0.434
	1	4	0.522	0.375	0.436
	2	2	0.629	0.450	0.525
strong nil match	1	1	0.067	0.020	0.031
	1	2	0.064	0.023	0.034
	1	3	0.067	0.020	0.031
	1	4	0.077	0.051	0.061
	2	2	0.142	0.055	0.080
strong all match	1	1	0.505	0.327	0.397
	1	2	0.492	0.319	0.387
	1	3	0.506	0.328	0.398
	1	4	0.454	0.325	0.379
	2	2	0.505	0.327	0.397
entity match	1	1	0.470	0.530	0.498
	1	2	0.456	0.506	0.480
	1	3	0.468	0.528	0.496
	1	4	0.456	0.506	0.480
	2	2	0.504	0.585	0.541
strong mention match	1	1	0.642	0.416	0.505
	1	2	0.642	0.416	0.505
	1	3	0.642	0.416	0.505
	1	4	0.606	0.435	0.506
	2	2	0.704	0.503	0.587

Table 1: Preliminary results for 4 runs in the first window and 1 run in the second window of the TAC 2016.

Model / Entity-Match	P	R	F
Model 1	0.517	0.623	0.565
Model 2	0.501	0.624	0.556
Model 3	0.520	0.621	0.566

Table 2: TAC 2016 entity match measure; Model 1: only mention model, Model 2: only context model (k=1), Model 3: only context model (K=6).

not encoded in our indexer because the heuristic method that encodes such relation has not seen such pairs in the corpus.

Position	P(Gold $\in$ Candidate-Set)
1-5	0.8669
6-10	0.0134
11-300	0.0213
$>$ 300 (Out-of-set)	0.0984

Table 3: Cumulative distribution of the relative position of the gold in the candidate set.

7 Conclusion

We participated in the cross lingual entity discovery and linking task at the TAC KBP 2016. Our proposed model consists of mention and context models. In the mention model we rely on lexical features that capture the dependencies among queries and entities. We use transliteration when featurizing the query mentions in a non-English language.
The context model focuses on enhancing the mention model without adding language-dependent features. The experimental results show that the contextual evidence extracted from the document is as capable as the lexical and heuristic evidence encoded in the mention model.
In encoding the contextual information into mention and entity embeddings we use skip-gram model. However in the future our plan is to use different approach especially to encode the hierarchical information of the entities and also the structure of the context into embeddings.

References

[Blanco *et al.*, 2015] Roi Blanco, Giuseppe Ottaviano, and Edgar Meij. Fast and space-efficient entity linking in queries. *WSDM*, 2015.

[Duchi *et al.*, 2011] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 2011.

[Durrett and Klein, 2015] Greg Durrett and Dan Klein. A joint model for entity analysis:coreference, typing, and linking. *NAACL*, 2015.

[Globerson *et al.*, 2016] Amir Globerson, Nevena LazicandSoumen Chakrabarti, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. Collective entity resolution with multi-focal attention. *ACL*, 2016.

[He *et al.*, 2013] Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. Learning entity representation for entity disambiguation. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL*, 2013.

[Lee *et al.*, 2011] Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Stanfords multi-pass sieve coreference resolution system at the conll-2011 shared task. *Shared Task, CONLL Shared Task 11, pages 2834, Stroudsburg, PA, USA. Association for Computational Linguistics*, 2011.

[Ma *et al.*, 2014] Chao Ma, Janardhan Rao Doppa, Walker Orr, Prashanth Mannem, Xiaoli Fern, Thomas G. Dietterich, and Prasad Tadepalli. Prune-and-score: Learning for greedy coreference resolution. *EMNLP*, 2014.

[Manning *et al.*, 2014] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. *ACL*, 2014.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *NIPs*, 2013.

[Stoyanov *et al.*, 2010] V. Stoyanov, C. Cardie, N. Gilbert, E. Riloff, D. Buttler, and D. Hysom. Coreference resolution with reconcile. *ACL*, 2010.

[Tsaia and Roth, 2016] Chen-Tse Tsaia and Dan Roth. Cross-lingual wikification using multilingual embeddings. *Proceedings of NAACL-HLT*, 2016.