Bridging Text and Knowledge by Learning Multi-Prototype Entity Mention Embedding

Yixin Cao[1], **Lifu Huang**[2], **Heng Ji**[2], **Xu Chen**[1], **Juanzi Li**[1*]
[1] Tsinghua National Laboratory for Information Science and Technology
Dept. of Computer Science and Technology, Tsinghua University, China 100084
{caoyixin2011,successcx,lijuanzi2008}@gmail.com
[2] Dept. of Computer Science, Rensselaer Polytechnic Institute, USA 12180
{huangl7,jih}@rpi.edu

Abstract

Integrating text and knowledge into a unified semantic space has attracted significant research interests recently. However, the ambiguity in the common space remains a challenge, namely that the same mention phrase usually refers to various entities. In this paper, to deal with the ambiguity of entity mentions, we propose a novel Multi-Prototype Mention Embedding model, which learns multiple sense embeddings for each mention by jointly modeling words from textual contexts and entities derived from a knowledge base. In addition, we further design an efficient language model based approach to disambiguate each mention to a specific sense. In experiments, both qualitative and quantitative analysis demonstrate the high quality of the word, entity and multi-prototype mention embeddings. Using entity linking as a study case, we apply our disambiguation method as well as the multi-prototype mention embeddings on the benchmark dataset, and achieve state-of-the-art performance.

1 Introduction

Jointly learning text and knowledge representations in a unified vector space greatly benefits many Natural Language Processing (NLP) tasks, such as knowledge graph completion (Han et al., 2016; Wang and Li, 2016), relation extraction (Weston et al., 2013), word sense disambiguation (Mancini et al., 2016), entity classification (Huang et al., 2017) and linking (Huang et al., 2015).

Existing work can be roughly divided into two categories. One is encoding words and entities into a unified vector space using Deep Neural Networks (DNN). These methods suffer from the problems of expensive training and great limitations on the size of word and entity vocabulary (Han et al., 2016; Toutanova et al., 2015; Wu et al., 2016). The other is to learn word and entity embeddings separately, and then align similar words and entities into a common space with the help of Wikipedia hyperlinks, so that they share similar representations (Wang et al., 2014; Yamada et al., 2016).

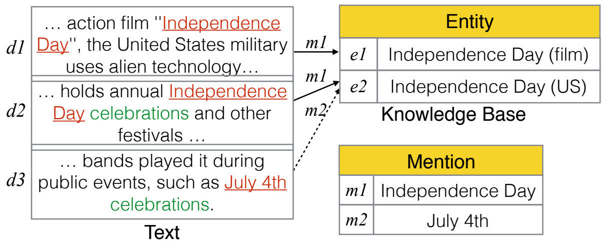

Figure 1: Examples.

However, there are two major problems arising from directly integrating word and entity embeddings into a unified semantic space. First, mention phrases are highly ambiguous and can refer to multiple entities in the common space. As shown in Figure 1, the same mention *independence day* (m_1) can either refer to a holiday: *Independence Day (US)* or a film: *Independence Day (film)*. Second, an entity often has various aliases when mentioned in various contexts, which implies a much larger size of mention vocabulary compared with entities. For example, in Figure 1, the documents d_2 and d_3 describes the same entity *Independence Day (US)* (e_2) with distinct mentions: *independence day* and *July 4th*. We observe tens of millions of mentions referring to 5 millions of entities in Wikipedia.

To address these issues, we propose to learn multiple embeddings for mentions inspired by the Word Sense Disambiguation (WSD) task (Reisinger and Mooney, 2010; Huang et al., 2012;

*Corresponding author.

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 1623–1633
Vancouver, Canada, July 30 - August 4, 2017. ©2017 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17-1149

Tian et al., 2014; Neelakantan et al., 2014; Li and Jurafsky, 2015). The basic idea behind it is to consider entities in KBs that can provide a meaning repository of mentions (i.e. words or phrases) in texts. That is, each mention has one or multiple meanings, namely **mention senses**, and each sense corresponds to an entity. Furthermore, we assume that different mentions referring to the same entity express the same meaning and share a common mention sense embedding, which largely reduces the size of mention vocabulary to be learned. For example, the mentions *Independence Day* in d_2 and *July 4th* in d_3 have a common mention sense embedding during training since they refer to the same holiday. Thus, text and knowledge are bridged via mention sense.

In this paper, we propose a novel **M**ulti-**P**rototype **M**ention **E**mbedding (MPME) model, which jointly learns the representations of words, entities, and mentions at sense level. Different mention senses are distinguished by taking advantage of both textual context information and knowledge of reference entities. Following the frameworks in (Wang et al., 2014; Yamada et al., 2016), we use separate models to learn the representations for words, entities and mentions, and further align them by a unified optimization objective. Extending from skip-gram model and CBOW model, our model can be trained efficiently (Mikolov et al., 2013a,b) from a large scale corpus. In addition, we also design a language model based approach to determine the sense for each mention in a document based on multi-prototype mention embeddings.

For evaluation, we first provide qualitative analysis to verify the effectiveness of MPME to bridge text and knowledge representations at the sense level. Then, separate tasks for words and entities show improvements by using our word, entity and mention representations. Finally, using entity linking as a case study, experimental results on the benchmark dataset demonstrate the effectiveness of our embedding model as well as the disambiguation method.

2 Preliminaries

In this section, we formally define the input and output of multi-prototype mention embedding.

A **knowledge base** $\mathcal{KB}$ contains a set of entities $\mathcal{E} = \{e_j\}$, and their relations. We use Wikipedia as the given knowledge base, and organize it as a directed knowledge network: nodes denote entities, and edges are outlinks from Wikipedia pages. In the directed network, we define the entities that point to e_j as its neighbors $\mathcal{N}(e_j)$, but ignore those entities that e_j points to, so that the repeated computations on the same edge would be avoided if edges were undirected.

A **text corpus** $\mathcal{D}$ is a set of sequential words $\mathcal{D} = \{w_1, \cdots, w_i, \cdots, w_{|\mathcal{D}|}\}$, where w_i is the ith word and $|\mathcal{D}|$ is the length of the word sequence. Since an entity mention m_l may consist of multiple words, we define an annotated text corpus[1] as $\mathcal{D}' = \{x_1, \cdots, x_i, \cdots, x_{|\mathcal{D}'|}\}$, where x_i corresponds to either a word w_i or a mention m_l. We define the words around x_i within a predefined window as its context words $\mathcal{C}(x_i)$.

An **Anchor** is a Wikipedia hyperlink from a mention m_l linking to its entity e_j, and is represented as a pair $< m_h, e_j > \in \mathcal{A}$. The anchors provide mention boundaries as well as their reference entities from Wikipedia articles. These Wikipedia articles are used as an annotated text corpus $\mathcal{D}'$ in this paper.

Multi-Prototype Mention Embedding . Given a $\mathcal{KB}$, an annotated text corpus $\mathcal{D}'$ and a set of anchors $\mathcal{A}$, we aim to learn multi-prototype mention embedding, namely multiple sense embeddings $\mathbf{s_j}^l \in \mathbb{R}^k$ for each mention m_l as well as word embeddings $\mathbf{w}$ and entity embeddings $\mathbf{e}$. We use $\mathcal{M}_l^* = \{s_j^l\}$ to denote the sense set of mention m_l, where each s_j^l refers to an entity e_j. Thus, the vocabulary size is reduced to a fixed number $|\{s_j^*\}| = |\mathcal{E}|$. We use s_j^* to denote the shared sense of mentions referring to entity e_j.

Example As shown in Figure 1, *Independence Day* (m_1) has two mention senses s_1^1, s_2^1, and *July 4th* (m_2) has one mention sense s_2^2. Based on the assumption in Section 1, we have $s_2^* = s_2^1 = s_2^2$ referring to entity *Independence Day (US)* (e_2).

3 An Overview of Our Method

Given a knowledge base $\mathcal{KB}$, an annotated text corpus $\mathcal{D}'$ and a set of anchors $\mathcal{A}$, we aim to jointly learn word, entity and mention sense representations: $\mathbf{w}$, $\mathbf{e}$, $\mathbf{s}$.

As shown in Figure 2, our framework contains two key components:

[1]Generally, the mention boundary can be obtained by using NER tools like Standford NER (Finkel et al., 2005). In this paper, we use Wikipedia anchors as annotations of Wikipedia text corpus for the concentration of our main purpose.

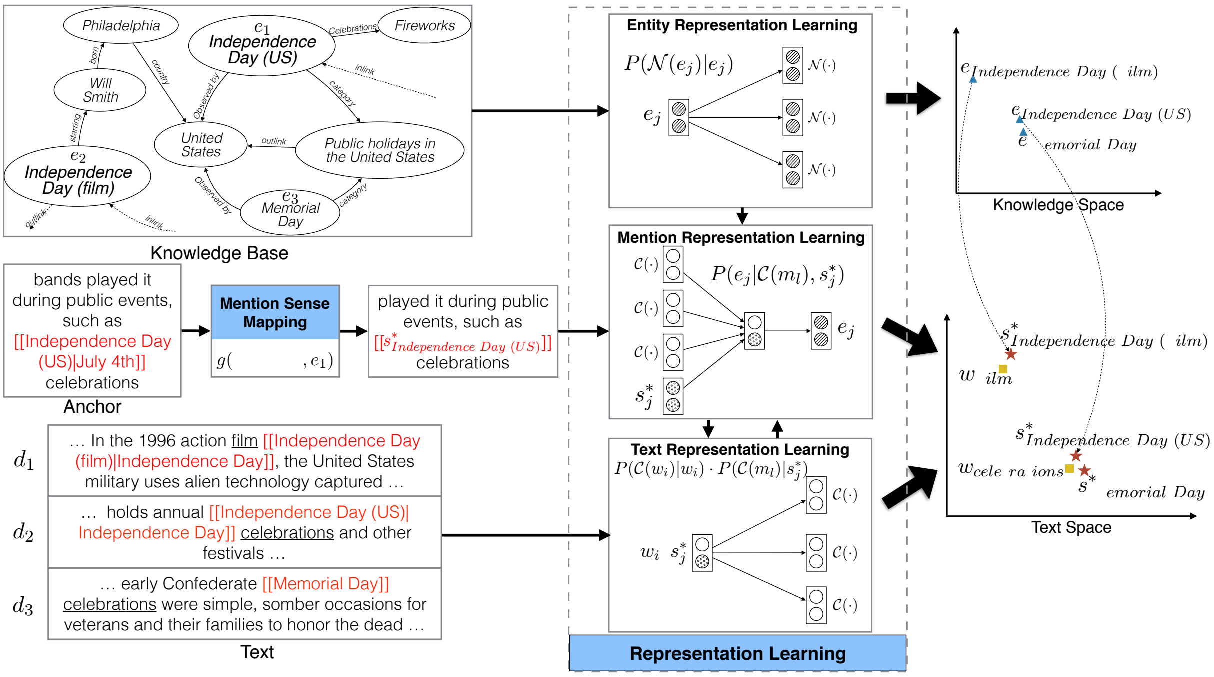

Figure 2: Framework of Multi-Prototype Mention Embedding model.

Mention Sense Mapping To reduce the size of the mention vocabulary, each mention is mapped to a set of shared mention senses according to a predefined dictionary. We build the dictionary by collecting entity-mention pairs $< m_l, e_j >$ from Wikipedia anchors and page titles, then create mention senses if there is a different entity. The sense number of a mention depends on how many different entity-mention pairs it is involved.

Formally, we have: $\mathcal{M}^*_l = g(m_l) = \bigcup g(< m_l, e_j >) = \{s^*_j\}$, where $g(\cdot)$ denotes the mapping function from an entity mention to its mention sense given an anchor. We directly use the anchors contained in the annotated text corpus D' for training. As Figure 2 shows, we replace the anchor <*July 4th*, *Independence Day (US)*> with the corresponding mention sense: $s^*_{Independence\ Day\ (US)}$.

Representation Learning Using $\mathcal{KB}$, $\mathcal{A}$ and $\mathcal{D}'$ as input, we design three separate models and a unified optimization objective to jointly learn entity, word and mention sense representations into two semantic spaces. As shown in the knowledge space in Figure 2, entity embeddings can reflect their relatedness in the network. For example, *Independence Day (US)* (e_1) and *Memorial Day* (e_3) are close to each other because they share some common neighbors, such as *United States* and *Public holidays in the United States*.

Word and mention embeddings are learned in the same semantic space. As two basic units in $\mathcal{D}'$, their embeddings represent their distributed semantics in texts. For example, mention *Independence Day* and word *celebrations* co-occur frequently when it refers to the holiday: *Independence Day (US)*, thus they have similar representations. Without disambiguating the mention senses, some words, such as *film* will also share similar representations as *Independence Day*.

Besides, by introducing entity embeddings into our MPME framework, the knowledge information will also be distilled into mention sense embeddings, so that the mention sense *Memorial Day* will be similar as *Independence Day (US)*.

Mention Sense Disambiguation According to our predefined dictionary, each mention has been mapped to more than one senses, and learned with multiple embedding vectors. Consequently, to induce the correct sense for a mention within a context is critical in the usage of the multiprototype embeddings, especially in an unsupervised way. Formally, given an annotated document $\mathcal{D}'$, we determine one sense $\hat{s}^*_j \in \mathcal{M}^*_l$ for each mention $m_l \in \mathcal{D}'$, where $\hat{s}^*_j$ is the correct sense.

Based on language model, we design a mention sense disambiguation method without using any supervision that takes into account three aspects: 1) **sense prior** denotes how dominant the sense is, 2) **local context information** reflects how semantically appropriate the sense is in the context, and

3) **global mention information** denotes how semantically consistent the sense is with the neighbor mentions. To better utilize the context information, we maintain a context cluster for each mention sense during training, which will be detailed in Section 4.4.

Since each mention sense corresponds to an entity in the given KB, the disambiguation method is equivalent to entity linking. Thus, text and knowledge base is bridged via the multiprototype mention embeddings. We will give more analysis in Section 6.4.

4 Representation Learning

Distributional representation learning plays an increasing important role in many fileds (Bengio et al., 2013; Zhang et al., 2017, 2016) due to its effectiveness for dimensionality reduction and addressing sparseness issue. For NLP tasks, this trends has been accelerated by the Skip-gram and CBOW models (Mikolov et al., 2013a,b) due to its efficiency and remarkable semantic compositionality of embedding vectors. In this section, we first briefly introduce the Skip-gram and CBOW models, and then extend them to three variants for the word, mention and entity representation learning.

4.1 Skip-Gram and CBOW model

The basic idea of the Skip-gram and CBOW models is to model the predictive relations among sequential words. Given a sequence of words $\mathcal{D}$, the optimization objective of Skip-gram model is to use the current word to predict its context words by maximizing the average log probability:

$$\mathcal{L} = \sum_{w_i \in \mathcal{D}} \sum_{w_o \in \mathcal{C}(w_i)} \log P(w_o | w_i) \tag{1}$$

In contrast, CBOW model aims to predict the current word given its context words:

$$\mathcal{L} = \sum_{w_i \in \mathcal{D}} \log P(w_i | \mathcal{C}(w_i)) \tag{2}$$

Formally, the conditional probability $P(w_o|w_i)$ is defined using a softmax function:

$$P(w_o|w_i) = \frac{\exp(\mathbf{w_i} \cdot \mathbf{w_o})}{\sum_{w_o \in \mathcal{D}} \exp(\mathbf{w_i} \cdot \mathbf{w_o})} \tag{3}$$

where $\mathbf{w_i}, \mathbf{w_o}$ denote the input and output word vectors during training. Furthermore, these two models can be accelerated by using hierarchical softmax or negative sampling (Mikolov et al., 2013a,b).

4.2 Entity Representation Learning

Given a knowledge base $\mathcal{KB}$, we aim to learn entity embeddings by modeling "contextual" entities, so that the entities sharing more common neighbors tend to have similar representations. Therefore, we extend Skip-gram model to a network by maximizing the log probability of being a neighbor entity.

$$\mathcal{L}_e = \sum_{e_j \in \mathcal{E}} \log P(\mathcal{N}(e_j)|e_j) \tag{4}$$

Clearly, the neighbor entities serve a similar role as the context words in Skip-gram model. As shown in Figure 2, entity *Memorial Day* (e_3) also share two common neighbors of *United States* and *Public holidays in the United States* with entity *Independence Day (US)*, thus their embeddings are close in the Knowledge Space. These entity embeddings will be later used to learn mention representations.

4.3 Mention Representation Learning

As mentioned above, the textual context information and reference entities are helpful to distinguish different senses for a mention. Thus, given an anchor $< m_l, e_j >$ and its context words $\mathcal{C}(m_l)$, we combine mention sense embeddings with its context word embeddings to predict the reference entity by extending CBOW model. The objective function is as follows:

$$\mathcal{L}_m = \sum_{<m_l, e_j> \in \mathcal{A}} \log P(e_j | \mathcal{C}(m_l), s_j^*) \tag{5}$$

where $s_j^* = g(< m_l, e_j >)$. Thus, if two mentions refer to similar entities and share similar contexts, they tend to be close in semantic vector space. Take Figure 1 as an example again, mentions *Independence Day* and *Memorial Day* refer to similar entities *Independence Day (US)* (e_1) and *Memorial Day* (e_2), they also share some similar context words, such as *celebrations* in documents d_2, d_3, so their sense embeddings are close to each other in the text space.

4.4 Text Representation Learning

Instead of directly using a word or a mention to predict the context words, we incorporate mention sense to joint optimize word and sense representations, which can avoid some noise introduced by ambiguous mentions. For example, in Figure 2, without identifying the mention *Independence Day* as the holiday or the film, various dissimilar context words such as the words *celebrations* and *film* in documents d_1, d_2 will share similar semantics, which will further affect the performance of entity representations during joint training.

Given the annotated corpus $\mathcal{D}'$, we use a word w_i or a mention sense s_j^* to predict the context words by maximizing the following objective function:

$$\mathcal{L}_w = \sum_{w_i, m_l \in \mathcal{D}'} \log P(\mathcal{C}(w_i)|w_i) + \log P(\mathcal{C}(m_l)|s_j^*) \tag{6}$$

where $s_j^* = g(< m_l, e_j >)$ is obtained from anchors in Wikipedia articles.

Thus, words and mention senses will share the same vector space, where similar words and mention senses are close to each other, such as *celebrations* and *Independence Day (US)* because they frequently occur in the same contexts.

Similar to WDS, we maintain a context cluster for each mention sense, which can be used for mention sense disambiguation (Section 5). The context cluster of a mention sense s_j^* contains all the context vectors of its mention m_l. We compute context vector of m_l by averaging the sum of its context word embeddings: $\frac{1}{|\mathcal{C}(m_l)|}\sum_{w_j \in \mathcal{C}(m_l)} \mathbf{w_j}$. Further, the center of a context cluster μ_j^* is defined as the average of context vectors of all mentions which refer to the sense. These context clusters will be later used to disambiguate the sense of a given mention with its contexts.

4.5 Joint Training

Considering all of the above representation learning components, we define the overall objective function as linear combinations:

$$\mathcal{L} = \mathcal{L}_w + \mathcal{L}_e + \mathcal{L}_m \tag{7}$$

The goal of training MPME is to maximize the above function, and iteratively update three types of embeddings. Also, we use negative sampling technique for efficiency (Mikolov et al., 2013a).

MPME shares the same entity representation learning method with (Yamada et al., 2016), but the role of entities in the entire framework as well as mention representation learning is different in three aspects. First, we focus on learning embeddings for mentions, not merely words as in (Yamada et al., 2016). Clearly, MPME is more natural to integrate text and knowledge base. Second, we propose to learn multiple embeddings for each mention denoting its different meanings. Third, we prefer to use both mentions and context words to predict entities, so that the distribution of entities will help improve word embeddings, meanwhile, avoid being hurt if we force entity embeddings to satisfy word embeddings during training (Wang et al., 2014). We will give more analysis in experiments.

5 Mention Sense Disambiguation

As mentioned in Section 3, we induce a correct sense $\hat{s}_j^* \in \mathcal{M}_l^*$ for each mention m_l in an annotated document $\mathcal{D}'$. We regard this problem from the perspective of language model that maximizes a joint probability of all mention senses contained in the document. However, the global optimum is expensive with a time complexity of O($|\mathcal{M}||\mathcal{M}_l^*|$). Thus, we approximately identify each mention sense independently:

$$\begin{aligned} & P(\mathcal{D}', \ldots, s_j^*, \ldots,) \\ \approx & \prod P(\mathcal{D}'|s_j^*) \cdot P(s_j^*) \\ \approx & \prod P(\mathcal{C}(m_l)|s_j^*) \cdot P(\hat{\mathcal{N}}(m_l)|s_j^*) \cdot P(s_j^*) \end{aligned} \tag{8}$$

where $P(\mathcal{C}(m_l)|s_j^*)$, local context information (Section 3), denotes the probability of the local contexts of m_l given its mention sense s_j^*. we define it proportional to the cosine similarity between the current context vector and the sense context cluster center μ_j^* as described in Section 4.4. It measures how likely a mention sense occurring together with current context words. For example, given the mention sense *Independence Day (film)*, word *film* is more likely to appear within the context than the word *celebrations*.

$P(\hat{\mathcal{N}}(m_l)|s_j^l)$, global mention information, denotes the probability of the contextual mentions of m_l given its sense s_j^l, where $\hat{\mathcal{N}}(m_l)$ is the collection of the neighbor mentions occurring together with m_l in a predefined context window. We define it proportional to the cosine similarity between mention sense embeddings and the neighbor mention vector, which is computed similar to context vector: $\sum \frac{1}{|\hat{\mathcal{N}}(m_l)|}\mathbf{\hat{s}^l_j}$, where $\hat{s}^l_j$ is the correct sense for m_l.

Considering there are usually multiple mentions in a document to be disambiguated. The mentions disambiguated first will be helpful for inducing the senses of the rest mentions. That is, how to choose the mentions disambiguated first will influence the performance. Intuitively, we adopt two orders similar to (Chen et al., 2014): 1) L2R (left to right) induces senses for all the mentions in the document following natural order that varies according to language, normally from left to right in the sequence. 2) S2C (simple to complex) denotes that we determine the correct sense for those mentions with fewer senses, which makes the problem easier.

Global mention information assumes that there should be consistent semantics in a context window, and measures whether all neighbor mentions are related. For instance, two mentions *Memorial Day* and *Independence Day* occur in the same document. If we already know that *Memorial Day* denotes a holiday, then obviously *Independence Day* has higher probability of being a holiday than a film.

$P(s^*_j)$, sense prior, is a prior probability of sense s^*_j indicating how possible it occurs without considering any additional information. We define it proportional to the frequency of sense s^*_j in Wikipedia anchors:

$$P(s^*_j) = (\frac{|\mathcal{A}_{s^*_j}|}{|\mathcal{A}|})^{\gamma} \quad \gamma \in [0, 1]$$

where $\mathcal{A}_{s^*_j}$ is the set of anchors annotated with s^*_j, and γ is a smoothing hyper-parameter to control the impact of prior on the overall probability, which is set by experiments (Section 6.4).

6 Experiment

Setup We choose Wikipedia, the March 2016 dump, as training corpus, which contains nearly 75 millions of anchors, 180 millions of edges among entities and 1.8 billions of tokens after preprocessing. We then train MPME[2] for 1.5 millions of words, 5 millions of entities and 1.7 millions of mentions. The entire training process in 10 iterations costs nearly 8 hours on the server with 64 core CPU and 188GB memory.

We use the default settings in word2vec[3], and set our embedding dimension as 200 and context window size as 5. For each positive example, we sample 5 negative examples[4].

Baseline Methods As far as we know, this is the first work to deal with mention ambiguity in the integration of text and knowledge representations, so there is no exact baselines for comparison. We use the method in (Yamada et al., 2016) as a baseline, marked as **ALIGN**[5], because (1) this is the most similar work that directly aligns word and entity embeddings. (2) it achieves the state-of-the-art performance in entity linking task.

To investigate the effect of multi-prototype, we degrade our method to single-prototype as another baseline, which means to use one sense to represent all mentions with the same phrase, namely **S**ingle-**P**rototype **M**ention **E**mbedding (SPME). For example, SPME only learns one unique sense vector for *Independence Day* whatever it denotes a holiday or a film.

6.1 Qualitative Analysis

We use cosine similarity to measure the similarity of two vectors, and present the top 5 nearest words and entities for two most popular senses of the mention *Independence Day*. Because ALIGN is incapable of dealing with multiple words, we only present the results of SPME and MPME.

As shown in Figure 1, without considering mention sense, the mention *Independence Day* can only show a dominant *holiday* sense based on SPME and ignore all other senses. Instead, MPME successfully learns two clear and distinct senses. For the sense *Independence Day (US)*, all of its nearest words and entities, such as *parades*, *celebrations*, and *Memorial Day*, are holiday related, while for another sense *Independence Day (film)*, its nearest words and entities, like *robocop* and *The Terminator*, are all science fiction films. The results verify the effectiveness of our framework in learning mention embeddings at the sense level.

[2]Our main code for MPME can be found in https://github.com/TaoMiner/bridgeGap.

[3]https://code.google.com/archive/p/word2vec/

[4]We tested different parameters (e.g. window size of 10 and dimension of 500) which achieve similar results, and report the current settings considering program runtime efficiency.

[5]We carefully re-implemented ALIGN and used the same shared parameters in our model for fairly comparison. However, we failed to fully reproduce the positive result in the original paper, meanwhile the authors are unable to release their code.

	Mention Sense	Nearest words	Nearest entities
SPME	Independence Day	lee-jackson, thanksgiving, diwali, strassenfest, chiraghan	National Aboriginal and Torres Strait Islander Education Policy, E. Chandrasekharan Nair, Jean Aileen Little, Thessalian barbel, 1825 in birding and ornithology
MPME	Independence Day (US)	thanksgiving, parades, lee-jackson, festivities, celebrations	Memorial Day, Labor Day, Thanksgiving, Thanksgiving (United States), Saint Patrick's Day
	Independence Day (film)	robocop, clockstoppers, mindhunters, tarantino, terminator	The Terminator, True Lies, Total Recall (1990 film), RoboCop 2, Die Hard

Table 1: The nearest neighbors of mention *Independence Day*.

6.2 Entity Relatedness

To evaluate the quality of entity embeddings, we conduct experiments using the dataset which is designed for measuring entity relatedness (Ceccarelli et al., 2013; Huang et al., 2015; Yamada et al., 2016). The dataset contains 3,314 entities, and each mention has 91 candidate entities on average with gold-standard labels indicating whether they are semantically related.

We compute cosine similarity between entity embeddings to measure their relatedness, and rank them in a descending order. To evaluate the ranking quality, we use two standard metrics: normalized discounted cumulative gain (NDCG) (Järvelin and Kekäläinen, 2002) and mean average precision (MAP) (Schütze, 2008).

We design another baseline method: **Entity2vec**, which learns entity embeddings using the method described in Section 4.2, without joint training with word and mention sense embeddings.

Table 2: Entity Relatedness.

	NDCG			MAP
	@1	@5	@10	
ALIGN	0.416	0.432	0.472	0.410
Entity2vec	0.593	0.595	0.636	0.566
SPME	0.593	0.594	0.636	0.566
MPME	**0.613**	**0.613**	**0.654**	**0.582**

As shown in Table 2, ALIGN achieves lower performance than Entity2vec, because it doesn't consider the mention phrase ambiguity and yields lots of noise when forcing entity embeddings to satisfy word embeddings and aligning them into the unified space. For example, the entity *Gente (magazine)* should be more relevant to the entity *France*, the place where its company locates. However, ALIGN mixed various meanings of mention *Gente* (e.g., the song) and ranked some bands higher (e.g., entity *Poolside (band)*).

SPME also doesn't consider the ambiguity of mentions but achieves comparative results with Entity2vec. We analyze the reasons and find that, it can avoid some noise by using word embeddings to predict entities. MPME outperforms all the other methods, which demonstrates that the unambiguous textual information is helpful to refine the entity embeddings.

6.3 Word Analogical Reasoning

Following (Mikolov et al., 2013a; Wang et al., 2014), we use the word analogical reasoning task to evaluate the quality of word embeddings. The dataset consists of 8,869 semantic questions (*"Paris":"France"::"Rome":?*), and 10,675 syntactic questions (e.g., *"sit":"sitting"::"walk":?*). We solve it by finding the closest word vector $\mathbf{w}_?$ to $\mathbf{w}_{France} - \mathbf{w}_{Paris} + \mathbf{w}_{Rome}$ according to cosine similarity. We compute accuracy for top 1 nearest word to measure the performance.

Table 3: Word Analogical Reasoning.

	Word2vec	ALIGN	SPME	MPME
Semantic	66.78	68.34	71.65	71.65
Syntactic	61.58	59.73	55.28	54.75

We also adopt Word2vec[6] as an additional baseline method, which provides a standard to measure the impact from other components on word embeddings.

Table 3 shows the results. We can see that ALIGN, SPME and MPME, achieve higher performance in dealing with semantic questions, because relations among entities (e.g., country-capital relation for entity *France* and *Paris*) enhance the semantics in word embeddings through jointly training. On the other hand, their performance for syntactic questions is weakened because more accurate semantics yields a bias to predict semantic relations even though given a syntactic query. For example, given the query *"pleasant":"unpleasant"::"possibly":?*, our

[6]https://code.google.com/archive/p/word2vec/

model tends to return the word (e.g., *probably*) highly semantical related to query words, such as *possibly*, instead of the syntactical similar word *impossibly*. In this scenario, we are more concerned about semantic task to incorporate knowledge of reference entities into word embeddings, and this issue could be tackled, to some extent, by using syntactic tool like stemming.

The word embeddings of MPME achieve the best performance for semantic questions mainly because (1) text representation learning has better generalization ability due to the larger size of training examples than entities (e.g., 1.8b v.s. 0.18b) as well as relatively smaller size of vocabulary (e.g., 1.5m v.s. 5m). (2) unambiguous mention embeddings capture both textual context information and knowledge, and thus enhance word and entity embeddings.

6.4 A Case Study: Entity Linking

Entity linking is a core NLP task of identifying the reference entity for mentions in texts. The main difficulty lies in the ambiguity of various entities sharing the same mention phrase. Previous work addressed this issue by taking advantage of the similarity between words and entities (Francis-Landau et al., 2016; Sun et al., 2015), and/or the relations among entities (Thien Huu Nguyen, 2016; Cao et al., 2015). Therefore, we use entity linking as a case study for a comprehensive measurement of the multi-prototype mention embeddings. Given mentions in a text, entity linking aims to link them to a predefined knowledge base. One of the main challenges in this task is the ambiguity of entity mentions.

We use the public dataset AIDA created by (Hoffart et al., 2011), which includes 1,393 documents and 27,816 mentions referring to Wikipedia entries. The dataset has been divided into 946, 216 and 231 documents for the purpose of training, developing and testing. Following (Pershina et al., 2015; Yamada et al., 2016), we use a publicly available dictionary to generate candidate entities and mention senses. For evaluation, we rank the candidate entities for each mention and report both standard micro (aggregates over all mentions) and macro (aggregates over all documents) precision over top-ranked entities.

Supervised Entity Linking

Yamada et al. (2016) designed a list of features for each mention and candidate entity pair. By incorporating these features into a supervised learning-to-rank algorithm, Gradient Boosting Regression Tree (GBRT), each pair is assigned a relevance score indicating whether they should be linked to each other. Following their recommended parameters, we set the number of trees as 10,000, the learning rate as 0.02 and the maximum depth of the decision tree as 4.

Based on word and entity embeddings learned by ALIGN, the key features in (Yamada et al., 2016) are from two aspects: (1) the cosine similarity between context words and candidate entity, and (2) the coherence among "contextual" entities in the same document.

To evaluate the performance of multi-prototype mention embeddings, we incorporate the following features into GBDT for comparison: (1) the cosine similarity between the current context vector and the sense context cluster center μ_j^*, which denotes how likely the mention sense refers to the candidate entity, (2) the cosine similarity between the current context vector and the mention sense embeddings.

Table 4: Performance of Supervised Method

	ALIGN	SPME	MPME
Micro P@1	0.828	0.820	**0.851**
Macro P@1	0.862	0.844	**0.881**

As shown in Table 4, we can see that ALIGN performs better than SPME. This is because SPME learns word embeddings and entity embeddings in separate semantic spaces, and fails to measure the similarity between context words and candidate entities. However, MPME computes the similarity between context words with mention sense instead of entities, thus achieves the best performance, which also demonstrates the high quality of the mention sense embeddings.

Unsupervised Entity Linking

Linking a mention to a specific entity equals to disambiguating mention senses since each candidate entity corresponds to a mention sense. As described in Section 5, we disambiguate senses in two orders: (1) **L2R** (from left to right), and (2) **S2C** (from simple to complex).

We evaluate our unsupervised disambiguation methods on the entire AIDA dataset. To be fair, we choose the state-of-the-art unsupervised methods, which are proposed in (Hoffart et al., 2011; Alhelbawy and Gaizauskas, 2014; Cucerzan, 2007;

Table 5: Performance of Unsupervised Methods

	Cucerzan	Kulkarni	Hoffart	Shirakawa	Alhelbawy	MPME (L2R)	MPME (S2C)
Micro P@1	0.510	0.729	0.818	0.823	0.842	0.882	**0.885**
Macro P@1	0.437	0.767	0.819	0.830	0.875	0.875	**0.890**

Kulkarni et al., 2009; Masumi Shirakawa and Nishio, 2011) using the same dataset.

Table 5 shows the results. We can see that our two methods outperform all other methods. MPME (L2R) is more efficient and easy to apply, while MPME (S2C) slightly outperforms it because the additional step of ranking mentions according to their candidates number guarantees a higher disambiguation performance for those simple mentions, which consequently help disambiguate those complex mentions through global mention information in Equation 8.

We analyze the results and observe a disambiguation bias to popular senses. For example, there are three mentions in the sentence "***Japan*** *began the defence of their* ***Asian Cup I*** *title with a lucky 2-1 win against* ***Syria*** *in a Group C championship match on Friday*", where the country name *Japan* and *Syria* actually denote their national football teams, while the football match name *Asian Cup I* has little ambiguity. Compared to the team, the sense of country occurs more frequently and has a dominant prior, which greatly affects the disambiguation. By incorporating local context information and global mention information, both the context words (e.g., *defence* or *match*) and the neighbor mentions (e.g., *Asian Cup I*) provide us enough clues to identify a soccer related mention sense instead of the country.

Influence of Smoothing Parameter As mentioned above, a mention sense may possess a dominant prior and greatly affect the disambiguation. So we introduce a smoothing parameter γ to control its importance to the overall probability. Figure 3 shows the linking accuracy under different values of γ on the dataset of AIDA. $\gamma = 0$ indicates we don't use any prior knowledge, and $\gamma = 1$ indicates the case without smoothing parameter.

We can see that both micro and macro accuracy decrease a lot if we don't use the parameter ($\gamma = 1$). Only using local and global probabilities for disambiguation ($\gamma = 0$) achieves a comparable performance when $\gamma = 0.05$, both accuracy reach their peaks, which is optimal and default value in our experiments.

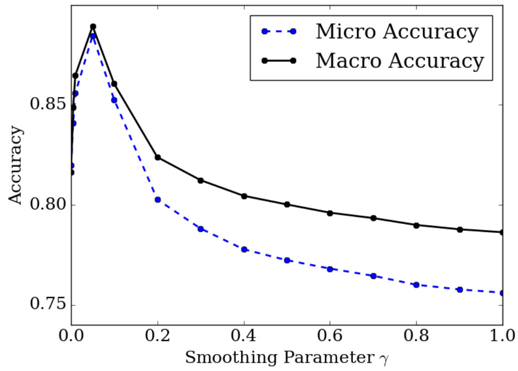

Figure 3: Impact of Smoothing Parameter γ.

7 Conclusions and Future Work

In this paper, we propose a novel Multi-Prototype Mention Embedding model that jointly learns word, entity and mention sense embeddings. These mention senses capture both textual context information and knowledge from reference entities, and provide an efficient approach to disambiguate mention sense in text. We conduct a series of experiments to demonstrate that multi-prototype mention embedding improves the quality of both word and entity representations. Using entity linking as a study case, we apply our disambiguation method as well as the multi-prototype mention embeddings on the benchmark dataset, and achieve the state-of-the-art.

In the future, we will improve the scalability of our model and learn multi-prototype embeddings for the mentions without reference entities in a knowledge base, and introduce compositional approaches to model the internal structures of multi-word mentions.

8 Acknowledgement

This work is supported by NSFC Key Program (No. 61533018), 973 Program (No. 2014CB340504), Fund of Online Education Research Center, Ministry of Education (No. 2016ZD102), Key Technologies Research and Development Program of China (No. 2014BAK04B03), NSFC-NRF (No. 61661146007) and the U.S. DARPA LORELEI Program No. HR0011-15-C-0115.

References

Ayman Alhelbawy and Robert J Gaizauskas. 2014. Graph ranking for collective named entity disambiguation. In *ACL (2)*. pages 75–80.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35(8):1798–1828.

Yixin Cao, Juanzi Li, Xiaofei Guo, Shuanhu Bai, Heng Ji, and Jie Tang. 2015. Name list only? target entity disambiguation in short texts. In *EMNLP*. pages 654–664. https://doi.org/10.18653/v1/D15-1077.

Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. 2013. Learning relatedness measures for entity linking. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, pages 139–148.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *EMNLP*. Citeseer, pages 1025–1035. https://doi.org/10.3115/v1/D14-1110.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data .

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, pages 363–370.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing semantic similarity for entity linking with convolutional neural networks. In *Proceedings of NAACL-HLT*. pages 1256–1261. https://doi.org/10.18653/v1/N16-1150.

Xu Han, Zhiyuan Liu, and Maosong Sun. 2016. Joint representation learning of text and knowledge for knowledge graph completion. *CoRR* abs/1611.04125.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 782–792.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proc. ACL*.

Hongzhao Huang, Larry Heck, and Heng Ji. 2015. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *arXiv preprint arXiv:1504.07678* .

Lifu Huang, Jonathan May, Xiaoman Pan, Heng Ji, Xiang Ren, Jiawei Han, Lin Zhao, and James A Hendler. 2017. Liberal entity extraction: Rapid construction of fine-grained entity typing systems. *Big Data* 5(1):19–31.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)* 20(4):422–446.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2009. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pages 457–466.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? In *Proc. EMNLP*. https://doi.org/10.18653/v1/D15-1200.

Massimiliano Mancini, José Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli. 2016. Embedding words and senses together via joint knowledge-enhanced training. *CoRR* abs/1612.02703.

Haixun Wang Yangqiu Song Zhongyuan Wang Kotaro Nakayama Takahiro Hara Masumi Shirakawa and Shojiro Nishio. 2011. Entity disambiguation based on a. technical report. In *Technical Report MSR-TR-2011-125*. Microsoft Research.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *NIPS*. pages 3111–3119.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proc. EMNLP*. https://doi.org/10.3115/v1/D14-1113.

Maria Pershina, Yifan He, and Ralph Grishman. 2015. Personalized page rank for named entity disambiguation. In *HLT-NAACL*. pages 238–243.

Joseph Reisinger and Raymond J Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Proc. NAACL*.

Hinrich Schütze. 2008. Introduction to information retrieval. In *Proceedings of the international communication of association for computing machinery conference*.

Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. In *IJCAI*. pages 1333–1339.

Nicolas Fauceglia Mariano Rodriguez-Muro Oktie Hassanzadeh Alfio Massimiliano Gliozzo Mohammad Sadoghi Thien Huu Nguyen. 2016. Joint learning of local and global features for entity linking via neural networks. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*. pages 2310–2320.

Fei Tian, Hanjun Dai, Jiang Bian, Bin Gao, Rui Zhang, Enhong Chen, and Tie-Yan Liu. 2014. A probabilistic model for learning multi-prototype word embeddings. In *COLING*.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. *ACL Association for Computational Linguistics* https://doi.org/10.18653/v1/D15-1174.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph and text jointly embedding. In *Proc. EMNLP*. https://doi.org/10.3115/v1/D14-1167.

Zhigang Wang and Juan-Zi Li. 2016. Text-enhanced representation learning for knowledge graph. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proc. ACL*.

Jiawei Wu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2016. Knowledge representation via joint learning of sequential text and knowledge graphs. *CoRR* .

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proc. CoNLL*. https://doi.org/10.18653/v1/K16-1025.

Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. 2017. Visual translation embedding network for visual relation detection. *arXiv preprint arXiv:1702.08319* .

Hanwang Zhang, Xindi Shang, Wenzhuo Yang, Huan Xu, Huanbo Luan, and Tat-Seng Chua. 2016. Online collaborative learning for open-vocabulary visual classifiers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 2809–2817.