Common Space Embedding of Primal-Dual Relation Semantic Spaces

Hidekazu Oiwa*
The University of Tokyo
Tokyo, Japan
hidekazu.oiwa@gmail.com

Jun'ichi Tsujii
Microsoft Research
Beijing, China
jtsujii@microsoft.com

Abstract

Explicit continuous vector representation such as vector representation of words, phrases, etc. has been proven effective for various NLP tasks. This paper proposes a novel method of constructing such vector representation for both entity-pairs and relation expressions which link them in text. Based on the insight of the duality of relations, the representation is constructed by embedding of two separately constructed semantic spaces, one for entity-pairs and the other for relation expressions, into a common semantic space. By representing the two different types of objects (i.e. entity-pairs and relation expressions) in the same semantic space, we can treat the two tasks, relation mining and relation expression mining (a.k.a. pattern mining), systematically and in a unified manner. The approach is the first attempt to construct a continuous vector representation for expressions whose validity can be explicitly checked by their proximities to known sets of entity-pairs. We also experimentally validate the effectiveness of the common space for relation mining and relation expression mining.

1 Introduction

Learning continuous vector representation for expressions which consist of more than one word has gained attention in recent years. Various representations have been constructed and used to measure semantic similarities between expressions in various tasks, such as analogical reasoning (Turney et al., 2003; Mikolov et al., 2013) and sentiment analysis (Turney and Littman, 2003; Socher et al., 2012). Many algorithms have been proposed to construct such continuous representations, depending on specific tasks in mind. In this paper, we propose a method for constructing a vector representation for binary relations, i.e., relations with two arguments. We demonstrate the effectiveness of the representation for relation mining and relation expression mining.

The method exploits the duality of a relation (Bollegala et al., 2010). While Bollegala et al. (2010) uses the duality in their co-clustering algorithm, we construct an explicit semantic space which reflects the two aspects of a given relation. We first construct two separate semantic spaces, one for pairs of named entities and another for relation expressions in text which link an entity-pair. A relation is supposed to correspond to a subset in each of these two spaces. The subset of entity-pairs is a set of pairs between which the relation holds. The subset is called the extension set of the relation. The subset of relation expressions consists a set of expressions which are used to link entity-pairs in the extension set.

The two semantic spaces are then embedded into a single common space. Figure 1 illustrates a brief summary of constructing a common semantic space. While the subsets which correspond to a specific relation are supposed to constitute natural clusters in the two original spaces, objects in the two spaces exchange useful information to each other and form a tighter cluster in the common space. Exchange of information takes place through common space embedding.

Since both entity-pairs and relation expressions have their vector representations in the common semantic space, one can easily enumerate relation expressions specific to a certain set of entity-pairs (re-

*This project was conducted while the first author stayed at Microsoft Research Asia.

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 1579–1590, Dublin, Ireland, August 23-29 2014.

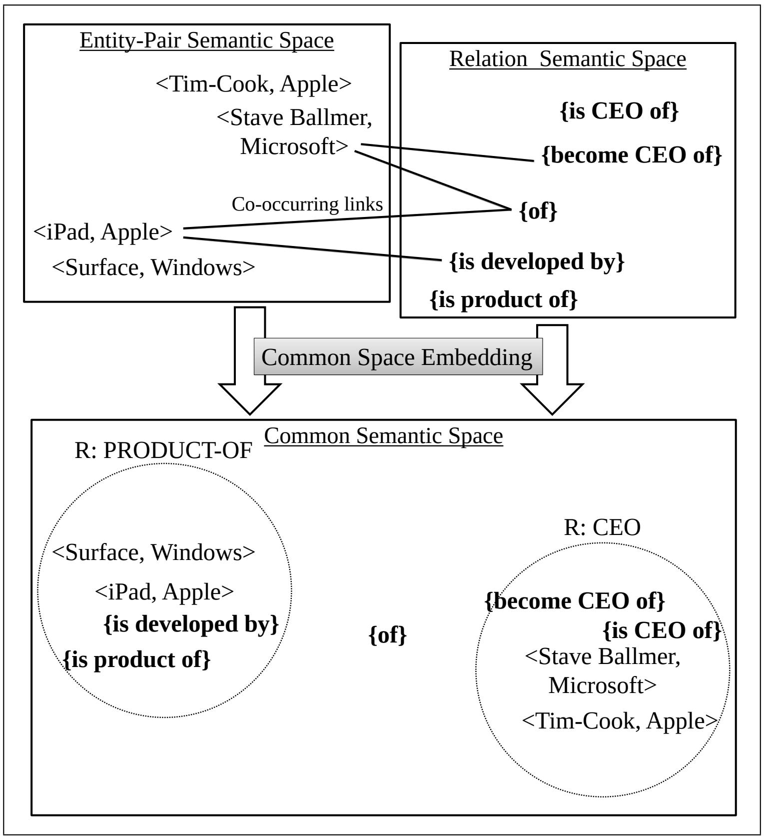

Figure 1: Overview of our framework to construct common semantic space.

lation expression mining). Furthermore, unlike the conventional pattern-based relation mining, one can perform relation mining in the common space without explicit reference to relation expressions.

2 Basic Framework

2.1 Duality of Relation and a Common Space

A binary relation is defined either extensionally by a set of pairs in the relation or intensionally by a set of conditions which a pair in the relation should satisfy. However, in actual applications of text mining, either of these definitions is given in a complete form. We are only given a subset of the whole set of pairs and have to complete the set (i.e. relation mining). Instead of an explicit intensional definition, we only have a set of observations in text where pairs in a relation are linked by certain linguistic expressions. Based on such observations, we have to judge whether a given pair holds the relation or not. Though some observed expressions are non-ambiguous and explicit for a relation (for example, "the birth place of A is B"), most of expressions are not (such as"A comes from B").

We call a set of pairs which define a relation as Extension set of a relation, while we call their observed expressions in text as Manifestation set. While these two sets are only partially given, they define relations which we are interested in. Such duality of a relation has been recognized by many previous work and has been exploited in relation mining and relation expression mining. (Bollegala et al., 2010), for example, used the duality in their work on co-clustering of entity-pairs and relation expressions. (Baroni and Lenci, 2010) presented a more general approach which defines a tensor associating a triplet $< e_1, l, e_2 >$ with a weight. e_1 and e_2 are entity pairs, while l is a linking expression in text. By projecting the tensor to matrices, they showed that diverse concepts used in distributional semantics could be captured in a unified manner. In particular, their tensors capture directly the duality of entity pairs and their linking expressions (i.e. relation expressions).

These previous works implicitly assume that the semantic space of entity pairs and that of relation expressions are tightly coupled. That is, the space of entity pairs is defined in terms of their co-occurrences with linking expressions (or the weights in a tensor between them) and vice versa. However, such tight coupling between the semantic spaces of entity pairs and relation expressions is not a logical necessity, and harmful in the sense that it restricts available information only to their co-occurrences.

An entity pair and a linking expression are complex objects by themselves, and their semantic spaces can be defined independently of each other. Two entities in a sentence, for example, are linked not only by single verbs or predicates but by a long sequence of words. This means that we can define a semantic space of linking expressions independently of entity pairs which they link. For example, one can use sequence similarities of words among relation expressions. Since knowledge resources of large scale have become available of late, we can define a semantic space of entity pairs by using paths in these knowledge graphs, regardless of their textual occurrences with relation expressions.

In this paper, we first define two separate semantic spaces (i.e. dual primal spaces) for entity pairs and relation expressions, and then use their textual co-occurrences to construct a common space consistent with the two primal spaces. In this approach, the co-occurrences of entity pairs with relation expressions play only an auxiliary role to project the two spaces into a common space.

The approach allows us to integrate information richer than mere co-occurrences of two objects (i.e. entity pairs and relation expressions). Furthermore, the common space provides us with direct means by which one can grasp finer grained relationships between two objects. Given a set of seed pairs of entities, one can gather a set of relation expressions in their nearest neighbor in the common space. Another set of seed pairs, even though conceptually they belong to the same relation, one may get a different set of relation expressions. The previous approaches, in which the semantics of the two objects are captured in two separate spaces, can capture only indirectly the hierarchical nature of natural relations, and how such a hierarchy is mapped on association of extension sets with manifestation sets.

2.2 Extension set and Manifestation Set

Let E be a set of named entities. Let $\langle e_i, e_j \rangle$ denote a pair of entities ($e_i, e_j \in E$) and E^2 a set of all entity-pairs. Then, a relation, R, is extensionally defined as a set of entity-pairs $E_R \subset E^2$, such as CEO = { ⟨Tim-Cook, Apple⟩, ⟨Ballmer, Microsoft⟩, ...}, COMPETE = {⟨Apple, Samsung⟩, ⟨Google, Microsoft⟩, ...} between which the relation holds. We call such a set of entity-pairs the extension set of a relation R.

On the other hand, a relation R is manifested in text in various forms of expressions. For example, "is the CEO of" in "Tim-Cook is the CEO of Apple" is a direct manifestation of the relation CEO. While "overtook" in "Samsung overtook Apple in the smartphone market in China" can be a manifestation of the relation COMPETE, this manifestation is rather indirect, based on inference. We denote a relation expression by r_i and the whole set of relation expressions by D. We call a subset of relation expressions which manifest, directly or indirectly, a relation R, as the manifestation set of R.

2.3 Primal-dual semantic spaces

A relation, R, is characterized by the two sets, the extension set and the manifestation set. In other words, the two sets are implicitly associated with each other via the relation R. This association between the two sets constitutes the foundation of the common semantic space to be constructed in this paper.

We first construct primal-dual semantic spaces, one for entity-pairs and another for relation expressions. A sentence where two entities appear can be seen from two different perspectives. One view is to see the sentence as characterization of the entity-pair, while the other takes the sentence as characterization of the relation expression which links the two entities. Based on these two views, we construct two semantic spaces from a given set of sentences (corpus). One space is for a set of entity-pairs (E^2) and the other for a set of relation expressions (D). $e^2 \in E^2$ and $r \in D$ are represented by vectors $\mathbf{e}^2 \in \mathbf{E}^2$ and $\mathbf{r} \in \mathbf{D}$ in the corresponding spaces. We assume that the two spaces are vector spaces, i.e., $\mathbf{E}^2$ and $\mathbf{D}$ are an n-dimensional vector space and an m-dimensional one, respectively.

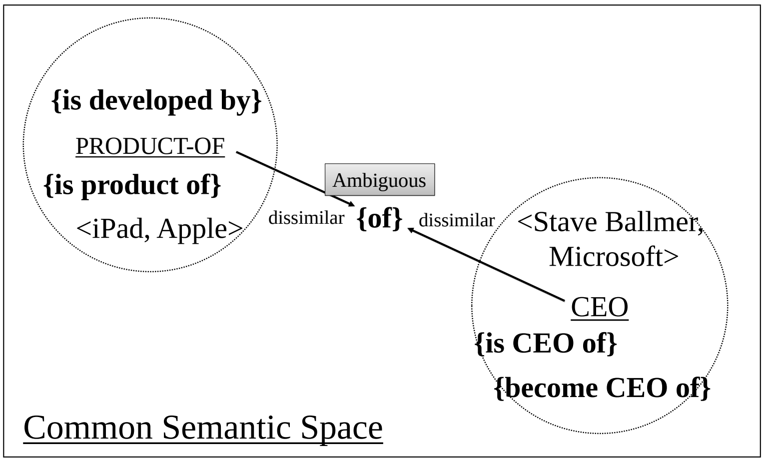

Figure 2: Illustration of common semantic space defined by our approach.

2.4 Triplets

The two objects, entity-pairs and relation expressions, whose spaces are separate, are linked through their co-occurrences in text. Co-occurrence of a relation expression (r) and an entity-pair ($e^2 = \langle e_1^2, e_2^2 \rangle$) means that r links in a sentence the entities of e_1^2 and e_2^2. A triplet represents such a co-occurrence with its frequency ($f \in \mathcal{R}$) in text. An instance of triplets is denoted as $\langle \mathbf{e}^2, \mathbf{r}, f \rangle \in T$. T indicates a set of triplets. These co-occurrence frequencies between entity-pairs and relation expressions play a critical role in common space embedding as linkage clues.

2.5 Common space embedding from $\mathbf{E}^2$ and $\mathbf{D}$

We use Multi-View Partial Least Squares (MVPLS) (Wu et al., 2013) as the basic framework to construct a common space from $\mathbf{E}^2 \subset \mathcal{R}^m$ and $\mathbf{D} \subset \mathcal{R}^n$. MVPLS was originally developed for web search and has been proven to be effective for embedding the semantic space of queries and that of documents into a common space. This framework is an extension of the conventional well-used approach, Partial Least Square. The framework is general enough to be used for our purpose.

Let k be the dimension of common latent space such that $k \leq m$ and $k \leq n$. $\mathbf{e}_i^2 \in \mathbf{E}^2$ is a i-th entity-pair feature vector in the entity-pair space and $\mathbf{r}_i \in \mathbf{D}$ is a i-th phrase feature vector. $\mathbf{L}_e$, $\mathbf{L}_r$ are linear projection matrices for embedding the original feature vector space into the common latent space. $\mathbf{L}_e$ is $m \times k$ and $\mathbf{L}_r$ is $n \times k$ size matrices.

MVPLS learns these two projection matrices for generating a well-constructed common space from the two separated spaces. Construction of latent common space can be formulated as an optimization problem which maximizes the sum of the similarities between entity-pairs and relation expressions in the common space when they co-occur. This optimization problem is as follows:

$$\underset{\mathbf{L}_e, \mathbf{L}_r}{\operatorname{argmax}} \sum_{(\mathbf{e}_i^2, \mathbf{r}_i, f_i) \in T} \log(f_i) \mathbf{r}_i^T \mathbf{L}_r \mathbf{L}_e^T \mathbf{e}_i^2 \quad s.t. \quad \mathbf{L}_e^T \mathbf{L}_e = \mathbf{I}, \quad \mathbf{L}_r^T \mathbf{L}_r = \mathbf{I} \,. \tag{1}$$

Note that the similarity score is weighted by the logarithmic scale of the co-occurrence counts. The outputs of this optimization problem are $\mathbf{L}_e$ and $\mathbf{L}_r$ which maximize the objective value where the orthogonal constraints on these matrices are satisfied. We do not necessarily solve (1) again when the system receives a new instance because the derived matrices can be applied not only for the existing entity-pairs and relation expressions but new ones. The problem is not convex, but Wu et al. (2013) proved that the global optimal solution can be obtained by SVD of $\sum_T \log(f_i) \mathbf{e}_i^2 \mathbf{r}_i^T$. $\mathbf{L}_e$ corresponds to left singular vectors and $\mathbf{L}_r$ consists of right singular vectors.

2.6 Ambiguity of Relation Expressions in the Common Space

Due to the ambiguity of relation expressions, the assumption that the manifestation set of the same R cluster around in proximity does not hold in reality. "of" in "Steve Ballmer of Microsoft" belongs to the manifestation set of CEO, while “of” in “iPad of Apple” belongs to the set of a different relation, PRODUCT-OF. Indirect manifestation such as “overtake” is another cause of ambiguity. Inference involved here is abductive in nature and not always valid. We may be able to infer COMPETE relation from “X overtake Y”, but “X overtake Y” can be a consequence of another relation such as COOPERATE.

Such an ambiguous expression belongs to the manifestation sets of more than one relation and thus would be located in a rather neutral position in the space. Since the common space reflects how frequently certain expressions are used to link entity-pairs, their positions in the space reflect the relative specificity to each relation cluster. Figure 2 illustrates how the ambiguity of a relation expression captured in the common space.

3 Relation Mining and Relation Expression Mining

In an actual situation, both the extension set and the manifestation set of a relation R are only partially known. To produce more comprehensive sets of these objects from large corpora is generally called mining. Two mining tasks have been studied so far, which are different, though mutually related.

We define relation mining as a task which, given a relation R, enumerates entity-pairs in the extension set. Another mining task (i.e. relation expression mining which is often performed as an auxiliary task of relation mining) is to gather a set of relation expressions which are manifestations of a given R.

3.1 Relation Mining

Relation mining is the task to enumerate entity-pairs of a relation R from a small given set of objects of a relation R. For example, if a set of relation expressions as the manifestation set of a relation R are given, one can produce a set of entity-pairs simply by identifying occurrences of relation expressions in text and producing the entity-pairs which are linked by them. Alternatively, if a small set of entity-pairs as a subset of the extension set of a relation R are given, one can produce a set of entity-pairs simply by gathering similar entity-pairs measured by relation expression co-occurrence vectors. These ideas have been shared by many mining systems called pattern-based relation mining systems.

The recall and precision of such a system are determined by the quality and quantity of the given set. If the given set is small, a system suffers low recall. On the other hand, if the set is large but contains many ‘ambiguous’ or ‘weak’ objects, a system suffers low precision.

Therefore, one of the keys for success of relation mining is how to gather a large initial set, which are effective, i.e. objects less ambiguous with high frequency. The common semantic space can be used not only to generate a comprehensive set but to measure the specificity of objects in terms of a given R, it also provides refined semantic measures between entity-pairs.

3.2 Relation Expression Mining

We have discussed semantic spaces of relation expressions and the common semantic space as if to define what constitutes a relation expression is straightforward. However, it is not trivial to define what constitutes a relation expression.

In the previous section, we treat “overtake” in “Apple overtook Samsung in the smart phone market” as a relation expression which manifests the relation “COMPETE”. However, one may argue that a pattern such as “X overtake Y in . . . market” should be treated as a basic unit of manifestation of the relation COMPETE. This longer expression is less ambiguous and thus more effective than the shorter pattern of “overtake”. On the other hand, the frequency of this pattern would be much less and thus less effective, compared with the shorter version. Mining of effective relation expressions (sometimes called “pattern mining”) has to address the problem of balancing the specificity and generality of relation expressions. Furthermore, one would like to identify the same relation expression in “Apple announced yesterday that it had overtaken Samsung which . . .” as in “Apple overtook Samsung in the smart phone market”.

In the experiments, we do not treat the process of pattern mining seriously. Instead, we used two conventional methods. The first method is to enumerate subsequences of words in the intervening part in a sentence between two entities, and use them as relation expressions. We expect less effective expressions as manifestation to be recognized in the common space. Another method is to use the shortest paths in dependency structures of sentences as relation expressions. Shortest paths can generalize surface variants of essentially the same relation expressions and reduce unnecessary proliferation of relation expressions.

4 Experiments

This section empirically evaluates our approach of embedding the two original spaces into a common space. We show that the common space provides a continuous vector space for relation expressions, in which not only similarities among expressions but also their ambiguities are properly captured.

4.1 Experiment Settings

4.1.1 Dataset

We use the ENT benchmark dataset (Bollegala et al., 2009) for our experiments. The dataset consists of 661,502 snippets, which are brief summaries provided by Web search engines. Most web search engines provide links to webpages and snippets as search results and snippets contains a subset of texts including the query words derived from the webpages. Table 1 shows how many distinct entity pairs, relation expressions and triplets were extracted as results of NER and expression extraction (See Section 4.1.2 and 4.1.3). The dataset is accompanied with 100 entity-pairs that are classified into five semantic categories: ACQUISITION, HEADQUARTERS, FIELD, CEO, and BIRTHPLACE. We use the ENT dataset not only for evaluation of relation mining but also for examining the characteristics of the common space for relation expression mining. Note that, due to the nature of snippets, the dataset is very noisy. It contains many non-sentences and even non-English texts, which may adversely affect the performance of mining systems.

	Entity-pair	Relation	Triplet
Enumeration	$12,174$	$12,185$	$521,454$
Shortest Path	$10,251$	$92,797$	$130,897$

Table 1: The specifications of the ENT dataset: Sizes of distinct entity-pairs, relation expressions, and triplets. "Enumeration" indicates the results of pattern mining based on word subsequences. "Shortest Path" shows that of shortest path extraction.

4.1.2 Entity and Entity-Pair Extraction

We first extracted entities from the ENT dataset. After splitting snippets into sentences, we applied named entity recognizer (NER) (Finkel et al., 2005) to recognize entities in sentences. We used Stanford Core NLP tools 2[1] for sentence splitting and NER. As relevant semantic classes for the ENT dataset, entities which are recognized as ORGANIZATION, LOCATION, or PERSON are treated as entities in the further process. We only used sentences in which at least two entities of these three classes appear.

4.1.3 Extraction of Relation Expressions

The definition of relation expressions which link two entities in text is not trivial. We adopt two methods of extracting candidates of relation expressions, and compare them in experiments.

The first method is to use, as relation expressions, subsequences of words which appear between two entities. We assume that two entities which appear apart in a sentence by more than 10 words are not explicitly linked in the sentence. From the word sequence whose length is less than 10, we enumerate all possible subsequences whose length is less than 6 words. Since a set of such subsequences include many noises as relation expressions, we use only subsequences the frequency of which is higher than 100.

This shallow approach can be run very fast, thanks to the advances of sequential pattern mining (Pei et al., 2004). Although the method is similar to that used in Bollegala et al. (2010) , we do not use any further constraints based on part-of-speech tags, lexical-syntactic information, etc. Our contention is that such ad-hoc constraints unnecessarily restrict a set of relation expressions. Our method treats ambiguous expressions (e.g. "of", "in", "with", etc.) as relation expressions. Instead, the effectiveness or the degree of ambiguities of a relation expression is captured in the common space after embedding.

[1] http://nlp.stanford.edu/software/corenlp.shtml

The second method is based on dependency parsing. We obtain the dependency tree of a sentence by a publicly available deep parser, Enju3[2] (Miyao and Tsujii, 2005; Miyao and Tsujii, 2008), and then extract shortest paths between two entities. Unlike the first method, this method uses linguistic information to extract the skeleton of a relation expression.

Each node in shortest paths consists of a base form (e.g., “like”, “player”), syntactic category (e.g., “verb”, “noun”), and predicate-argument links. The length of shortest paths was restricted to the range from 1 to 6. Compared with the first method, a set of shortest paths contains much less noises, so that we do not filter out those with low frequency. In the same way as the first method, a set of shortest paths contains highly ambiguous paths (e.g. the path of “of”).

4.1.4 Generation of the Space for Entity-Pairs

The primal semantic space for entity pairs can be constructed in several ways. The co-training method constructed a space of entity pairs based on their co-occurrences with relation expressions. Their method requires the two spaces of entity pairs and relation expressions have to be tightly coupled.

On the other hand, our approach allows us to design the two spaces independently. In addition to the tightly coupled spaces, we design a new space for entity pairs based on the distributional hypothesis (Harris, 1954). We used the point-wise mutual information (PMI) score of each word with an entity-pair. PMI score is defined as $\mathrm{PMI} = \log_e p(w_a|\langle e_i, e_j\rangle)/p(w_a)$ where $p(w_a)$ is an occurrence probability of a word w_a and $p(w_a|\langle e_i, e_j\rangle)$ is a conditional probability with respect to an entity-pair $\langle e_i, e_j\rangle$. We filtered words whose PMI scores were below 1.0 and all the rest were used as the features.

To maximize the effectiveness of the space, we performed preliminary experiments by changing parameters in the definition of context in the distributional hypothesis, such as how the context around entities is distinguished, whether the whole of a sentence or limited windows around entities are used as context, etc. As a result, we chose the settings in which right, left, and intervening contexts are distinguished. We used three different window sizes as the context (e.g. 4, 5 and 6 words). That is, when we set the window size to 4, we used the four words in the left side of the first entity as the left context, those in the right side of the second entity as the right context, and the words in the intervening part as the intervening context. If the intervening part consists of more than 8 words, the four words in the neighborhood of the two entities are used as the intervening context.

4.1.5 Generation of the Space for Relation Expressions

Following the work (Lin and Pantel, 2001), we constructed a simple space, in which a relation expression is characterized by the entities which it links. We counted the entities in the left-hand side and the right hand side of a relation expression. The same as the vector of an entity-pair, we used the PMI score as the feature value. As for feature selection, we chose the entities whose PMI scores are no less than 1.0[3].

4.1.6 Dimension Reduction

After generating vectors for entity-pairs and relation expressions, we applied a dimension reduction. Since both of the primary semantic spaces use surface words or entities, their vectors tend to have a very large dimension (i.e. about $100,000$ for entity pairs and about $2,500$ for relation expressions). Since the cardinalities of the two sets of distinct entity pairs and relations expressions are also very high (See Table 1 of the specification of the ENT dataset), the high dimensions of the two spaces would make the computation cost of MVPLS embedding in terms of time and space prohibitively high.

To take advantage of the sparseness of both spaces, we used Randomized SVD (Halko et al., 2011) which can produce low-dimensional feature vectors from a large-scale sparse feature matrix efficiently. We produced spaces with $3,000$-dimensions for entity-pairs and $1,000$ for relation expressions.

4.1.7 Common Space Embedding

Lastly, we applied MVPLS (1) to construct common space projection matrices. We set the dimension of common space as $1,000$. We verified that the dimension does not affect much the evaluation results,

[2] `http://www.nactem.ac.uk/enju/`

[3] Other than context-based characterization methods, we have applied path kernel method (Reichartz et al., 2009; Reichartz et al., 2010) to shortest path relations as preliminary works, however, their performances were definitely worse.

Window Size	4	5	6
VSM (Turney, 2005)		0.68	
LRA (Turney, 2005)		0.68	
(Bollegala et al., 2010)		0.76	
Relation $(1,000)$		0.82	
Original $(1,000)$	0.88	0.88	0.88
Embedded $(1,000)$	$\mathbf{0.90}$	$\mathbf{0.89}$	$\mathbf{0.90}$

Table 2: Entity-Pair space evaluation results (Enumeration) : Each figure shows the average precision. The best figures in each window size are written in **bold**. Figures in parentheses denote the number of dimensions.

Window Size	4	5	6
VSM (Turney, 2005)		0.68	
LRA (Turney, 2005)		0.68	
(Bollegala et al., 2010)		0.76	
Relation $(1,000)$		0.62	
Original $(1,000)$	$\mathbf{0.91}$	0.90	$\mathbf{0.91}$
Embedded $(1,000)$	$\mathbf{0.91}$	$\mathbf{0.91}$	$\mathbf{0.91}$

Table 3: Entity-Pair space evaluation results (Shortest Path). Each figure shows the average precision. The best figures in each window size are written in **bold**. Figures in parentheses denote the number of dimensions.

when we set it to larger than 300. So we used a common space with $1,000$ dimensions for the sake of comparison with the original spaces.

4.2 Relation Mining Evaluation

We evaluated the embedding approach by a quantitative analysis on the relation mining task used in (Bollegala et al., 2010). The experimental setting is the same as the previous work. The objective is to assess whether the derived common semantic space provides a good space for measuring semantic distances among entity-pairs. We expected that in a good semantic space, entity-pairs which belong to the same semantic category would be clustered in proximity.

We used the ENT dataset (Bollegala et al., 2009). We used the same evaluation measures used in (Bollegala et al., 2010). The measure assumes that a semantic space would be judged as appropriate if it assigned higher similarity scores to entity-pairs the relationships of which belong to the same category. Therefore, the measure evaluated the top 10 similar pairs to each entity-pair and calculated average precision defined as $\sum_{t=1}^{10} \mathrm{Rel}(t) \cdot \mathrm{Pre}(t)/10$. Here, $\mathrm{Rel}(t)$ is a binary valued function that returns 1 if the entity-pair at rank t and $\langle e_i, e_j \rangle$ have the same semantic category. $\mathrm{Pre}(t)$ is the precision at rank t, which is defined by the percentage of correct objects in top t pairs.

For the sake of comparison, we prepared several models, which used different semantic spaces for entity pairs. One space (called Relation) is to characterize an entity pair by the relation expressions which it co-occur. Another space (called Original) is to characterize an entity pair by the context vector discussed in Section 4.1.4. There are three Original spaces which use different window sizes (4, 5 and 6 words). Then, the final space is the common space obtained by embedding (called Embedded).

Table 2 and 3 correspond to the experiment results using the two definitions of relation expressions, one by enumerated word sequences and the other by shortest paths. We note that the previous works only use co-occurrences information and cannot use any context information. The previous work and Relation have no ways of changing the size of windows. Therefore, these results are independent of the window size. These tables show the limitation of co-training which can only use tightly coupled vector spaces for entity pairs and relation expressions. Both the original and the common embedded space outperform significantly the performance obtained by previous works, regardless of the definitions of relation expressions (i.e. enumerated subsequence and shortest paths). Since the space for relation expressions is simple and poor, we expected that it would hardly add extra information to the space of entity pairs. However, the common space embedded from the two spaces improve the performance.

4.3 Relation Expression Mining

While the primary space for relation expressions is rather poor, vector representations of relation expressions are much richer in the common space. This is because they receive extra information from the rich space of entity-pairs through their co-occurrences. For evaluation, we first chose representative relation expressions, and then gathered relations that are close to them in the primary space of relation

{announce acquisition}		{president ,}	
Embedded	Original	Embedded	Original
{announce that have acquire}	{announce that have acquire}	{chairman ,}	{'s president be}
{complete acquisition}	{acquire}	{, ceo &}	{would say}
{say have it buy}	{pay}	{'s president ,}	{would that say}
{acquire}	{buy}	{ceo &}	{'s blue and}
{pay}	{compra}	{chief ,}	{'s chairman ,}
{'s acquisition}	{buy company}	{, ceo)}	{chairman ,}
{'s out_of}	{say that it buy}	{chief ,}	{palmisano}
{'s purchase}	{nor}	{would that say}	{,}
{acquisition}	{do}	{executive ,}	{, reader ,}
{'s takeover}	{announce be buy}	{ceo become}	{palmisano include door '}

Table 4: Evaluation of similarity measure between relation expressions. This table shows the top-10 ranked relation expressions that are closest to two representative relation expressions.

expressions and in the common space. If our expectation was correct, the list of expressions close to the chosen expression in the common space should be more appropriate than that in the primary space.

We show the result of the experiment in which we use shortest paths as relation expressions. We used the same dataset as the previous experiment. We removed shortest paths with frequency less than 10. As for the primary space for entity pairs, we use the one with the window size of 5. We used {announce acquisition} and {president ,} as two representatives.

Table 4 shows the lists of relation expressions closest to the chosen representatives in the common space and the primary space. For the ease of interpretation, we do not show syntactic categories and predicates attached to the shortest paths. One can easily see that the common space successfully moved down many ambiguous expressions such as {compra} and {nor} in {announce acquisition}, and {would say} and{,} in {president ,}. On the other hand, some relation expressions which are specific and semantically similar to the chosen ones moved up in the rank, for example {'s purchase} and {chief ,}.

We have also conducted the same experiment for relation expressions produced by the enumeration method. While the enumeration method improves the relation mining which gathering similar entity-pairs, it gave much poorer results to expressing mining than the shortest paths. This is because the enumeration method generated a large amount of non-meaningful relation expressions. For example, to generate a complex relation expression such as {say have it buy} appeared in Table 4, the enumeration method has to generate a large variety of noisy ones that co-occur with a complex relation expression.

4.4 Similarity measure between entity-pair and relation expressions

The major advantage of embedding over co-training is that it produces where the two different types of objects, entity-pairs and relation expressions, are treated in the exactly the same vector space. Therefore, we can easily gather a set of relation expressions relevant to a given prototype entity pair of a relation. In this experiment, instead of representative relation expressions, we gave entity pairs which are prototypical examples of certain relations. As in the previous experiment, we used the shortest paths as relation expressions, and ignored relation expressions with frequency under 10.

Table 5 shows the list of relation expressions for two prototypical entity-pairs used in the ENT dataset, ⟨charlie chaplin, london⟩ as a representative entity-pair for BIRTHPLACE and ⟨facebook inc, mark zuckerberg⟩ as CEO relation semantics. The table shows that the top-10 frequently co-occurring relation expressions. While many noisy relation expressions (i.e. ambiguous expressions) appear by extracting expressions based on their co-occurrence frequency with ⟨charlie chaplin, london⟩, these ambiguous expressions disappear in the proximity of the entity-pair in the common space. Moreover, the result of ⟨facebook inc. mark zuckerberg⟩ shows that some relation expressions that do not co-occur with the prototype entity-pair were successfully extracted, such as {'s executive ,}.

5 Related Work

Bollegala et al. (2010) proposed a simple sequential co-clustering framework of entity-pairs and relation expressions for objects sharing the same semantic relation to be clustered. Our definition of primal-dual

⟨charlie chaplin, london⟩		⟨facebook inc, mark zuckerberg⟩	
Embedded	Co-occurrence	Embedded	Co-occurrence
{bear walworth}	{bear}	{'s executive ,}	{, ceo}
{bear april}	{'s " arrangement while lay orchestra}	{ceo be}	{, ceo (}
{play}	{,}	{ceo}	{founder and}
{bear}	{reception}	{,}	{everything , ceo}
{bear}	{'s}	{'s president ,}	{andceo}
{bear woolsthorpe ,}	{'s}	{have say}	{ceo ,}
{bear woolthrope}	{be when}	{, ceo ,}	{-}
{be member parliament}	{and}	{, ceo}	N/A
{bear woolsthorpe}	{bear april street , walworth ,}	{ceo become}	N/A
{'s}	{walk ,}	{buy}	N/A

Table 5: Relation expressions gathered by prototype entity-pairs on the ENT dataset. This table shows the top-10 ranked relation expressions that are closest to the representative entity-pairs ⟨charlie chaplin, london⟩ as BIRTHPLACE and ⟨facebook inc, mark zuckerberg⟩ as CEO. ⟨facebook inc, mark zuckerberg⟩ co-occurred with only seven discrete relation expressions.

semantic space and common space embedding approach can be viewed as extensions of their work by introducing feature spaces as characterizations. This extension enables to utilize each space's characterizations and calculate similarity between different types of objects. Baroni and Lenci (2010) proposed a framework that analyze triplets as a third-order tensor, called "distributional memory". By matricizing the tensor to second-order tensors, that is matrices, this framework can utilize the relationship between entity-pairs and relation expressions. They also propose the procedure for generating continuous vector representations of entities and relation expressions through the tensor decomposition techniques. However, this framework cannot use semantic spaces independently defined, therefore it is difficult to incorporate the similarity information between entity-pairs or similarities between relation expressions into the decomposition procedure in contract to our framework based on MVPLS. Lin and Pantel (2001) proposed a weakly supervised framework of mining paraphrases based on shortest paths as basic units to be mined. Our work can be viewed as an extension by mixing entity-pair characterizations with the extended distributional hypothesis by embedding.

Many other previous work have been proposed to construct a knowledge base, including relation expressions (Carlson et al., 2010; Fader et al., 2011; Nakashole et al., 2012). However, they cannot interactively predict semantic meanings of objects through labeled objects of the other space.

As for treatment of ambiguity, some previous work has focused on triplet clustering to disambiguate each triplet object known as relation extraction. Unlike other mining tasks, this task requires a system to disambiguate the meaning of a relation expression r in $\langle r, e_1, e_2\rangle$ which appears in a specific context. We did not treat this task in this paper, however, our framework would discharge the burden by showing the insight of ambiguities of each relation expression and entity-pair. Yao et al. (2011; 2012) proposed a new triplet clustering method through a generative probabilistic model. The model used surrounding contexts as features in both a sentence and document level to identify the meaning of each triplet. They demonstrated the effectiveness of their models compared with USP (Poon and Domingos, 2009) or DIRT (Lin and Pantel, 2001). Min et al. (2012) provided a simple and scalable triplet clustering algorithm in an unsupervised way and enables to incorporate various resources about entity and relation expressions. Chen et al. (2006) proposed a label propagation algorithm for relation extraction as a semi-supervised learning method by utilizing the information of parsing.

6 Conclusion

We propose a common space embedding framework which constructs a semantic space in which both entity-pairs and relation expressions are represented. We showed that our framework is effective to construct the extension set and the manifestation set of a relation R in this space. The results of experiments showed that the common space is further refined for tasks such as relation and relation expression mining, compared with the original two spaces. Moreover, we showed relation expressions collected from a small set of entity-pairs through the common space, which share the same semantics as being relevant.

There are several interesting future topics:

- how to iteratively collect objects from a dual object, like bootstrapping
- how to reduce surface diversities of relation expressions which are not abstracted away by simple method or shortest paths (by using methods such as SOL Pattern Model (Nakashole et al., 2012))
- How to combine a ground truth and non-textual knowledge stored in knowledge bases for characterizing entity-pairs with our framework
- How to extend the framework in order to deal with n-ary relations

References

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

Danushka T. Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2009. Measuring the similarity between implicit semantic relations from the web. In *Proc. of WWW*.

Danushka T. Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2010. Relational duality: unsupervised extraction of semantic relations between entities on the web. In *Proc. of WWW*.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proc. of AAAI*.

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. 2006. Relation extraction using label propagation based semi-supervised learning. In *Proc. of ACL*.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proc. of EMNLP*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proc. of ACL*.

Nathan Halko, Per G. Martinsson, and Joel A. Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Dekang Lin and Patrick Pantel. 2001. Dirt - discovery of inference rules from text. In *Proc. of KDD*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*.

Bonan Min, Shuming Shi, Ralph Grishman, and Chin-Yew Lin. 2012. Ensemble semantics for large-scale unsupervised relation extraction. In *Proc. of EMNLP-CoNLL*.

Yusuke Miyao and Jun'ichi Tsujii. 2005. Probabilistic disambiguation models for wide-coverage hpsg parsing. In *Proc. of ACL*.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic hpsg parsing. *Computational Linguistics*, 34(1):35–80.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. Patty: A taxonomy of relational patterns with semantic types. In *Proc. of EMNLP-CoNLL*.

Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. 2004. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1424–1440.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *Proc. of EMNLP*.

Frank Reichartz, Hannes Korte, and Gerhard Paass. 2009. Dependency tree kernels for relation extraction from natural language text. In *Proc. of ECML/PKDD (2)*.

Frank Reichartz, Hannes Korte, and Gerhard Paass. 2010. Semantic relation extraction with kernels over typed dependency trees. In *Proc. of KDD*.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP-CoNLL*, pages 1201–1211.

Peter D. Turney and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346.

Peter D. Turney, Michael L. Littman, Jeffrey Bigham, and Victor Shnayder. 2003. Combining independent modules to solve multiple-choice synonym and analogy problems. In *RANLP*, pages 482–489.

Peter D. Turney. 2005. Measuring semantic similarity by latent relational analysis. In *IJCAI*, pages 1136–1141.

Wei Wu, Hang Li, and Jun Xu. 2013. Learning query and document similarities from click-through bipartite graph with metadata. In *Proc. of WSDM*.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proc. of EMNLP*.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012. Unsupervised relation discovery with sense disambiguation. In *Proc. of ACL*.