Disambiguating Spatial Prepositions Using Deep Convolutional Networks

Kaveh Hassani, Won-Sook Lee
School of Electrical Engineering and Computer Science, University of Ottawa, Canada
kaveh.hassani@uottawa.ca and wslee@uottawa.ca

Abstract
We address the coarse-grained disambiguation of the spatial prepositions as the first step towards spatial role labeling using deep learning models. We propose a hybrid feature of word embeddings and linguistic features, and compare its performance against a set of linguistic features, pre-trained word embeddings, and corpus-trained embeddings using seven classical machine learning classifiers and two deep learning models. We also compile a dataset of 43,129 sample sentences from Pattern Dictionary of English Prepositions (PDEP). The comprehensive experimental results suggest that the combination of the hybrid feature and a convolutional neural network outperforms state-of-the-art methods and reaches the accuracy of 94.21% and F1-score of 0.9398.

Introduction
Evolution has shaped complex visual-spatial processing capabilities such as object recognition, object search and navigation through space almost in all advanced species. It has also equipped humans with the discriminative ability of expressing and communicating spatial knowledge through language (Landau and Jackendoff 1993). One of the ambitious goals of Artificial Intelligence is to simulate this cognitive process to improve the applications of spatial knowledge. These applications include human-robot interactions, natural language interfaces, machine vision, text-to-scene conversion systems, geographical information systems, question answering systems, search engines, and spatial databases. This process entails various underlying sub-processes such as detecting, representing, grounding, planning, and inferring the spatial relations. The first step in simulating this process is to detect the spatial signals embedded within a given utterance.

In English, these signals can employ various syntactic categories such as verbs, adverbs, adjectives, nouns, pronouns, and prepositions. Nevertheless, most of the spatial relations in English are canonically lexicalized as prepositions or prepositional phrases. The spatial prepositions appear in locative expressions which convey information about the spatial configuration of two or more objects in some space, and consist of a preposition, its objects, and the subject that the prepositional phrase modifies (Herskovits 1985). The object of the preposition is called landmark or relatum and the subject is called trajectory or locatum. A locative expression can be formalized as a predicate-argument structure. For instance, an utterance such as the laptop is on the desk can be represented as on(laptop, desk). At first sight, the formalization process appears relatively straightforward using the following heuristic. The word with preposition Part-Of-Speech (POS) tag within the prepositional phrase is the predicate, the noun phrase occurring before the preposition is the relatum, and the noun phrase following the preposition is the locatum. However, beside the challenges of locating the relatum and locatum (i.e., co-references, compound prepositions, compound locative expressions, etc.), it is not a trivial task to decide whether the preposition conveys spatial information.

Spatial prepositions can be classified into locative and directional prepositions. Locative prepositions describe the position of a located object in relation to a set of reference objects whereas directional prepositions describe a change of position or direction of the located object. Locative prepositions are categorized to projective and topological prepositions. Projective prepositions such as above, in front of, and to the left of stipulate the information regarding the direction of a located object in respect to a reference object whereas topological prepositions such as in, on, and near convey information about the topological arrangements among the objects, and can be further classified to simple topological prepositions such as in and on and proximity topological prepositions such as near and far from (Coventry and Garrod 2004).

Polysemy (i.e., capacity of a word to have multiple semantically relevant but distinct senses) is a common feature of the prepositions. The task of deciding the sense of a word in a given context is called word sense disambiguation and is considered as an AI-complete problem (Navigli...
Spatial prepositions usually show high polysemy and can have up to 25 different senses (e.g. upon with 8 spatial and 17 non-spatial senses) (Litkowski and Hargraves 2007; Tratz and Hovy 2009). We address the coarse-grained disambiguation of the spatial prepositions as the first step towards spatial role labeling. Given a preposition within a sentence, the goal is to decide whether the preposition has a spatial sense in that context. For example, none of the following examples adapted from (Dittrich et al. 2015; Kordjamshidi, Otterlo, and Moens 2011) carry spatial signals. Examples (2) and (5) are idioms referring to getting old and feeling ill, and the rest of the examples have either abstract relatum or locatum.

(1) She is always in my heart.
(2) Peter is over the hill.
(3) The senator is at the far left of the political spectrum.
(4) The thought in the back of my mind.
(5) She felt under the weather.

In this paper, we propose a hybrid feature of word embeddings and linguistic features, and compare its performance against a set of linguistic features, pre-trained word embeddings, and corpus-trained embeddings using seven classical machine learning classifiers and two deep learning embeddings, and corpus-trained embeddings using seven linguistic features, pre-trained word embeddings, and linguistic features, pre-trained universal word embeddings, and the hybrid features. We use four sets of features including the engineered linguistic features, pre-trained universal word embeddings, corpus-trained word embeddings, and the hybrid features.

Linguistic Features

The linguistic features consist of twelve engineered features including five lexical features, four syntactic features and three semantic features. For a given symmetric window \( W: [w_i, w_{i+k}, w_{i+k+1}, w_{i+k+2},..., w_i, w_{i+k}] \) of size \( 2k+1 \) centered on the preposition (i.e., \( k \) words to the left and \( k \) words to the right of the preposition), these features are as follows.

Lexical Features

(1) The unigrams:
\[ U: [w_{i+1},...,w_i,...,w_{i+k}] \]
(2) The lemmas:
\[ L: [l_{i+1},...,l_i,...,l_{i+k}] \]
(3) The 1-skip-bigrams:
\[ B: [w_i, w_{i+k}, w_{i+k+1}, w_{i+k+2},..., w_i, w_{i+k}] \]
(4) The probability of unigrams occurring in spatial examples:
\[ PU: [p(s(w_{i+k})), p(s(w_i)),..., p(s(w_{i+k}))] \]
(5) The probability of 1-skip-bigrams occurring in spatial examples:
\[ PB: [p(s(w_{i+k+1})), p(s(w_{i+k})),..., p(s(w_{i+k}))] \]

Syntactic Features

(6) The word level POS-tags of the words:
\[ POS_w: [pos_w(w_{i+k}),..., pos_w(w_i),..., pos_w(w_{i+k})] \]
(7) The phrase level POS-tags of the words:
\[ POS_p: [pos_p(w_{i+k}),..., pos_p(w_i),..., pos_p(w_{i+k})] \]
(8) The phrase level POS-tags of the immediate ancestors of the words:
\[ POS_a: [pos_a(w_{i+k}),..., pos_a(w_i),..., pos_a(w_{i+k})] \]

A few works are carried out on SemEval 2007 dataset consisting of 25,000 sample sentences and 32 prepositions (Litkowski and Hargraves 2007). Ye and Baldwin (2007) employed a set of features consisting of POS-tags, WordNet synsets, and named entities on context windows of 7 words, and reached the F1-score of 0.861. Hovy, Tratz, and Hovy (2010) used seven rules to select a set of discriminative words from the given sentence and then used them to extract 17 features as the inputs to a maximum entropy classifier. They reached the accuracy of 91.8% on coarse-grained classification task. As far as our knowledge is concerned, this is the first work that applies deep learning models to disambiguate the spatial prepositions.

Features
(9) The minimum syntactic distance between the words and the preposition in the parse tree:

\[ DI: [\text{dis}(w_i,k, w_j), \ldots, \text{0}, \ldots, \text{dis}(w_i, w_{i+k})] \]

**Semantic Features**

(10) The named entities of the words:

\[ NE: [\text{ne}(w_i), \ldots, \text{ne}(w_j), \ldots, \text{ne}(w_{i+k})] \]

(11) An indicator to decide whether a word is abstract or physical:

\[ PH: [\text{phy}(w_i), \ldots, \text{phy}(w_j), \ldots, \text{phy}(w_{i+k})] \]

(12) The collapsed dependencies between the preposition and the words:

\[ DE: [\text{dep}(w_i,k, w_i), \ldots, \text{Null}, \ldots, \text{dep}(w_i, w_i)] \]

Stanford CoreNLP toolkit (Manning et al. 2014) is utilized to extract features (1), (2), (6)-(10) and (12), and NLTK toolkit (Loper and Bird 2002) is used to extract feature (3). For a given pair of a preposition and a word \((p, w)\) feature (9) is computed as follows.

\[
\text{dist}(p, w) = d(p, r) + d(w, r) - 2 \times d(lca(p, w), r)
\]

\( r \) denotes the root of the parse tree, \( d(w, r) \) is the distance between the root and the word, and \( lca(w_i, w_j) \) is the lowest common ancestor of words \( w_i \) and \( w_j \). Feature (11) is computed as follows. First, all nouns are extracted and searched in WordNet. Those nouns that are found and are the children of the \( \text{physical object} \) synset (i.e., the hypernym hierarchy of at least one of their senses includes \( \text{physical object} \)) are marked as physical. Those nouns whose hypernym hierarchy of all the senses includes abstract entity are tagged as abstract. Those words that are not found in the WordNet are tagged as none. Also, to augment this feature, a pairwise logical \( \text{OR} \) is performed on the proper nouns captured by Named Entity Recognition (NER) in feature (10) and the results from WordNet. Finally, the linguistic feature \( \text{FL} \) is defined as the concatenation of the features (1)-(12) as follows.

\[
\text{FL}: [\text{ULPBUPBPOSW}, \text{POSP}, \text{POSA}, \text{DI}, \text{NE}, \text{PH}, \text{DE}]^T
\]

**Universal Word Embeddings**

Word embeddings are dense vector representation of the words learned in an unsupervised manner (Bengio et al. 2003). They can capture fine-grained semantic and syntactic regularities using vector arithmetic and reflect similarities and dissimilarities between the words (Pennington, Socher, and Manning 2014). Similar to vector space models such as latent semantic analysis (LSA) (Deerwester et al. 1990), word embeddings are based on the distributional hypothesis (i.e., the meaning of a word can be determined by looking at its context) (Harris 1954). However, instead of global matrix factorization methods such as singular value decomposition (SVD), word embeddings are learned based on the neural language models in which the word vector is the internal representation of the word within the network. Because word embeddings are learned using shallow networks, learning them is much faster than the matrix factorization methods (Levy and Goldberg 2014; Levy, Goldberg, and Dagan 2015).

Several models such as continuous Bag-of-Words (CBOW) and skip-gram with negative-sampling (SGNS) (also known as Word2Vec) (Mikolov, Chen et al. 2013; Mikolov, Sutskever et al. 2013), vector log-bilinear models (vLBL and ivLBL) (Luong, Socher, and Manning 2013), explicit word embeddings based on positive point-wise mutual information (PPMI) metric (Levy and Goldberg 2014), and global vectors for word representation (GloVe) (Pennington, Socher, and Manning 2014) are proposed in literature. It has been shown that if these models are trained on very large corpuses, the resulted vectors are universal word features that can be applied to various tasks (Kim 2014). In this study, we use pre-trained Word2Vec and GloVe word embeddings.

Word2Vec embeddings are trained using skip-gram with negative-sampling model on the local contexts (Mikolov, Chen et al. 2013; Mikolov, Sutskever et al. 2013). The Word2Vec pre-trained vectors are trained on part of Google News dataset with about 100 billion tokens. The model contains 300-dimensional vectors for 3 million words and phrases and is publicly available. Given the window \( W: [w_{i-k}, \ldots, w_{i}, \ldots, w_{i+k}] \), we define the Word2Vec representation of the window as follows.

\[ \text{W2Vemb}: [H_{w2v}(w_{i-k}), \ldots, H_{w2v}(w_i), \ldots, H_{w2v}(w_{i+k})] \]

\( H_{w2v}(w_k) \) is a hash function that retrieves the pre-trained word2vec vector of a given word \( w_k \). GloVe model explicitly utilize the word-context co-occurrence matrix. It leverages the combination of the statistical information of the global co-occurrence matrix with a weighted least squares regression model (Pennington, Socher, and Manning 2014). The GloVe embeddings are trained on a corpus collected by Common Crawl with 840 billion tokens. The model contains 300-dimensional vectors for 2.2 million words which is publicly available. Similarly, given the window \( W \), the GloVe feature is computed as follows.

\[ \text{Gemb}: [H_{G}(w_{i-k}), \ldots, H_{G}(w_i), \ldots, H_{G}(w_{i+k})] \]

**Local Word Embeddings**

In addition to pre-trained word2vec vectors, we trained the local word embeddings on a corpus of prepositions with 1,549,492 tokens based on SGNS model and using genism semantic modeling library (Rehurek and Sojka 2010). The model contains 50-dimensional and 100-dimensional vectors for 59,814 words. For a window \( W \), these local word embeddings are defined as follows.

\[ \text{L-W2Vemb}: [H_{w2v-l}(w_{i-k}), \ldots, H_{w2v-l}(w_i), \ldots, H_{w2v-l}(w_{i+k})] \]
Hybrid Features
We define the hybrid features as the amalgamation of the universal word embeddings and the linguistic features. For this purpose, we replace the linguistic features (1)-(5) with the corresponding word embeddings. As a result, the hybrid features is defined as follows.

\[ F_{\text{hy}}: [\text{Gemb POSW POSP POSA DI NE PH DE}]^T \]

Learning Models
As far as our knowledge is concerned, there is no solid baseline on the task of disambiguating the spatial prepositions. Hence, we utilize both classical machine learning models (seven models) and deep learning models (two models) for this task. The applied machine learning algorithms include a logistic regression classifier, a k-nearest neighbor classifier, a Bernoulli naïve Bayes classifier, a linear support vector machines (SVM), and three ensemble classifiers including a random forest classifier, an Ada-boost classifier, and a bagging classifier. These classifiers are applied using scikit-learn machine learning library (Pedregosa et al. 2011). We also use two deep learning models including a fully connected feedforward deep neural network (DNN) and a deep convolutional neural network (CNN). These deep models are implemented using TensorFlow library (Martín Abadi et al. 2015).

Dataset
The Pattern Dictionary of English Prepositions (PDEP) (Litkowski 2014) is a publicly available lexical resource collected as a part of The Preposition Project (TPP) for the study of preposition behavior. It contains example sentences drawn from three corpora including FrameNet, Oxford English Corpus, and British National Corpus. PDEP contains 82,329 annotated example sentences of 1,061 senses under 304 prepositions. The prepositions are classified into twelve classes including activity, agent, backdrop, cause, membership, exception, scalar, spatial, tandem, temporal, topic, and tributary preposition classes. It provides the senses of a preposition, its pattern, substitutable prepositions, syntactic position, semantic class, super-sense, cluster, relation, and primary implicature.

The spatial class contains 169 senses under 78 spatial prepositions and 19,413 positive annotated examples. To compile the dataset, we extracted all members of the spatial class (i.e., prepositions that have at least one spatial sense). We also filtered out 36 spatial prepositions including the archaic prepositions (e.g., betwixt, nigh and thwart), technical prepositions (e.g., abait), limited dialect prepositions (e.g., outwith), and prepositions with less than ten examples (e.g. fore and sans). This resulted in a corpus of 73 spatial prepositions and 596 senses out of which 169 are spatial. Among them, upon with 25 (8 spatial and 17 non-spatial) different senses has the highest polysemy. Above and midst are the only prepositions whose all senses are spatial, and by has the minimum ratio of spatial senses (2 spatial and 21 non-spatial). The corpus consists of 19,103 positive instances and 24,026 negative instances. The final dataset is an almost balanced dataset consisting of 43,129 samples with skewness of 0.2289 and Kurtosis measure of -1.9473. We split the dataset into a train set with 39,000 samples and a test set with 4,129 samples (2,000 spatial and 2,129 non-spatial samples).

Experimental Setup
Preprocessing
A few regular expressions are used to replace the numbers with \(<\text{NUM}>\) symbol, pad the context windows by \(<\text{PAD}>\) symbol, and remove all the characters except the alphabetic characters and the punctuation marks. The preprocessing of features is as follows. For the universal word embeddings, the missing words in the pre-trained models are randomly generated by sampling each dimension from \(U \sim [-1, +1]\). We also reduce the dimensionality of the pre-trained vectors from 300 to 10, 50, 100, and 200 using principal component analysis (PCA) to investigate the effects of different dimension on the training. For the word embeddings trained on the corpus, we do not perform any feature preprocessing. For the linguistic features, the numeric identifiers of the POS-tags, unigrams, 1-skip-bigrams, dependencies, named entities alongside with the probability vectors are mapped to the range of \([-1, +1]\) using min-max normalization. For each feature set, five context windows of sizes 3, 5, 11, 15, and 21 are considered (e.g. window size of 11 contains 5 words to the right and 5 words to the left of the preposition).

Model Setup
Both deep learning models are trained using Adam stochastic optimizer (Kingma and Ba 2015) with the learning rate of 1E-4 over the mini-batches of size 250. The mini-batches are uniformly sampled with replacement. Both models utilize a cross-entropy loss function with one-hot output representation. They also use dropout regularization (Srivastava et al. 2014) with probability of \(p=0.5\) and batch normalization (Ioffe and Szegedy 2015) on the input layer. Both models also utilize the rectified linear units (ReLU) in their hidden layers and a softmax function in their output layer. The DNN model is defined as a four layer network (3 hidden layers + softmax layer). The CNN (Figure 1) consists of two convolutional layers each followed by a max pooling layer and two fully connected layers.
We fixed the number of epochs to 50,000, and used 32, 64, and 128 filters with sizes of 2×5, 3×5, and 5×5 with stride of 1. We also used windows of sizes 1×3, 2×2, 3×2 and 3×5 for the pooling. The sizes of the hidden layers are considered as hyper-parameters and are optimized using the random search. For this purpose, 10% of the training set is sampled as the dev set. For different feature sets, these sizes are set to 500, 800, and 1,000.

The random search is also employed to optimize the hyper-parameters of the classical machine learning models. The number of neighbors in the K-NN classifier is set to 10. The number of estimators and the learning rate in the Ada-boost classifier are set to 50 and 0.9, respectively. For the random forest classifier, the number of estimators is set to 15 and the Gini index is used to decide the split. The number of maximum features for each split is set to the root square of the size of the feature. The number of estimators of the bagging classifier is also set to 15. The penalty parameter of the SVM models is set to 0.98.

Results and Discussion

We first investigate the effects of the context window size and the dimensions of the word embeddings on the F1-score. For the DNN model with different feature sets, these effects are shown in Figure 2 and Figure 3, respectively. The other models show similar effects as well. As illustrated in Figure 2, as the window size increases, the F1-score decrease proportionally. This suggests that the senses of a preposition only depend on a small local context window. That is because the dependent and the governor of a preposition which provide the clues about its sense tend to appear very close to the preposition (i.e., preceding and proceeding noun phrases). As shown, the best F1-score is achieved with the window size of 5 which implies that smaller window size misses useful information whereas larger windows introduce higher noise to signal ratio. Also, as shown in Figure 3, the best F1-score is achieved by reducing the dimensionality from 100. It is noteworthy that the optimal dimension size depends on the corpus size.

Second, we analyze the accuracy and F1-score of different learning models with respect to the exploited feature set. The results for the context window of size 5 are shown in Table 1. The following observations can be pointed out:

1) The pre-trained universal embeddings outperform the corpus-trained embeddings in terms of accuracy and F1-score by 4.85% and 5.39%, respectively. This implies that the PDEP corpus is not big enough to be directly used to train the word embeddings.

2) The GloVe pre-trained embeddings slightly outperform the Word2Vec pre-trained embeddings when used with classic classifiers. On the other hand, Word2Vec pre-trained embeddings achieves better F1-score when used with DNN. Nevertheless, as shown in Figures 2 and 3, the pre-trained features demonstrate similar behaviors.

3) The feature sets that exploit the corpus information (i.e., linguistic and hybrid features) outperform the universal features. Also, it is shown that combining information from both local corpus and universal word distributions results in the best performance.

5) Regardless of the classifier type, the hybrid feature outperforms other feature sets in both accuracy and F1-score. Because the universal embeddings convey information regarding the similarities and dissimilarities among words, integrating the universal distributions of the words with the local corpus information enhances the performance.

6) On average both deep learning models outperform the classic classifiers. And among the classic classifiers ensemble models outperform the rest.
We addressed the coarse-grained disambiguation of the spatial prepositions (spatial vs. non-spatial) as the first step towards spatial role labeling using deep learning models. For this purpose, we first proposed a hybrid feature which is the combination of the universal word embeddings and the linguistic features of the corpus. Also, we compiled a spatial dataset of 43,129 instances from TTPE database.

We compared the performance of the hybrid feature against a set of linguistic features, pre-trained word embeddings, and corpus-trained embeddings using seven classical machine learning classifiers and two deep learning models.

The results suggested that the context window size of 5 yields in better results. The results also suggested that regardless of the classifier type, the proposed hybrid feature achieves better results in comparison with other features. Finally, the results revealed that feeding the hybrid feature to a convolutional neural network with two convolution layers achieves the accuracy of 94.21% and the and F1-score of 0.9398 which is an improvement of 7.06% over (Kordjamshidi, Van Otterlo, and Moens 2011), which is significant at \( p < 0.001 \).

For this purpose, we first proposed a hybrid feature which is the combination of the universal word embeddings and the linguistic features of the corpus. Also, we compiled a spatial dataset of 43,129 instances from TTPE database.

We compared the performance of the hybrid feature against a set of linguistic features, pre-trained word embeddings, and corpus-trained embeddings using seven classical machine learning classifiers and two deep learning models.

The results suggested that the context window size of 5 yields in better results. The results also suggested that regardless of the classifier type, the proposed hybrid feature achieves better results in comparison with other features. Finally, the results revealed that feeding the hybrid feature to a convolutional neural network with two convolution layers achieves the accuracy of 94.21% and the and F1-score of 0.9398 which is an improvement of 7.06% over (Kordjamshidi, Van Otterlo, and Moens 2011), which is significant at \( p < 0.001 \).

As future work, we are planning to investigate the fine-grained spatial disambiguation. Also, we are planning to apply our approach to the general task of preposition sense disambiguation using the PDEP database (i.e. 82,329 examples). Finally, we are planning to apply the recurrent neural networks for joint disambiguating, and locating the prepositions, relatums, and locatums to address the spatial role labeling task.

**References**


