# Semantic Representations for Domain Adaptation: A Case Study on the Tree Kernel-based Method for Relation Extraction

**Thien Huu Nguyen**[†], **Barbara Plank**[§] **and Ralph Grishman**[†]
[†] Computer Science Department, New York University, New York, NY 10003, USA
[§] Center for Language Technology, University of Copenhagen, Denmark
thien@cs.nyu.edu,bplank@cst.dk,grishman@cs.nyu.edu

## Abstract

We study the application of word embeddings to generate semantic representations for the domain adaptation problem of relation extraction (RE) in the tree kernel-based method. We systematically evaluate various techniques to generate the semantic representations and demonstrate that they are effective to improve the generalization performance of a tree kernel-based relation extractor across domains (up to 7% relative improvement). In addition, we compare the tree kernel-based and the feature-based method for RE in a compatible way, on the same resources and settings, to gain insights into which kind of system is more robust to domain changes. Our results and error analysis shows that the tree kernel-based method outperforms the feature-based approach.

## 1 Introduction

Relation Extraction (RE) is an important aspect of information extraction that aims to discover the semantic relationships between two entity mentions appearing in the same sentence. Previous research on RE has followed either the kernel-based approach (Zelenko et al., 2003; Bunescu and Mooney, 2005; Zhao and Grishman, 2005; Zhang et al., 2006; Bunescu, 2007; Qian et al., 2008; Nguyen et al., 2009) or the feature-based approach (Kambhatla, 2004; Grishman et al., 2005; Zhou et al., 2005; Jiang and Zhai, 2007a; Chan and Roth, 2010; Sun et al., 2011). Usually, in such supervised machine learning systems, it is assumed that the training data and the data to which the RE system is applied to are sampled independently and identically from the same distribution. This assumption is often violated in reality and exemplified in the fact that the performance of the traditional RE techniques degrades significantly in such a domain mismatch case (Plank and Moschitti, 2013). To alleviate this performance loss, we need to resort to domain adaptation (DA) techniques to adapt a system trained on some *source domain* to perform well on new *target domains*. We here focus on the *unsupervised domain adaptation* (i.e., no labeled target data) and *single-system* DA (Petrov and McDonald, 2012; Plank and Moschitti, 2013), i.e., building a single system that is able to cope with different, yet related target domains.

While DA has been investigated extensively in the last decade for various natural language processing (NLP) tasks, the examination of DA for RE is only very recent. To the best of our knowledge, there have been only three studies on DA for RE (Plank and Moschitti, 2013; Nguyen and Grishman, 2014; Nguyen et al., 2014). Of these, Nguyen et al. (2014) follow the supervised DA paradigm and assume some labeled data in the target domains. In contrast, Plank and Moschitti (2013) and Nguyen and Grishman (2014) work on the unsupervised DA. In our view, unsupervised DA is more challenging, but more realistic and practical for RE as we usually do not know which target domains we need to work on in advance, thus cannot expect to possess labeled data of the target domains. Our current work therefore focuses on the single-system *unsupervised* DA. Besides, note that this setting tries to construct a single system that can work robustly with different but related domains (multiple target domains), thus being different from most previous studies on DA (Blitzer et al., 2006; Blitzer et al., 2007) which have attempted to design a specialized system for every specific target domain.

Plank and Moschitti (2013) propose to embed word clusters and latent semantic analysis (LSA) of words into tree kernels for DA of RE, while Nguyen and Grishman (2014) studies the appli-

cation of word clusters and word embeddings for DA of RE on the feature-based method. Although word clusters (Brown et al., 1992) have been employed by both studies to improve the performance of relation extractors across domains, the application of word embeddings (Bengio et al., 2003; Mnih and Hinton, 2008; Turian et al., 2010) for DA of RE is only examined in the feature-based method and never explored in the tree kernel-based method so far, giving rise to the first question we want to address in this paper:

*(i) Can word embeddings help the tree kernel-based methods on DA for RE and more importantly, in which way can we do it effectively?*

This question is important as word embeddings are real valued vectors, while the tree kernel-based methods rely on the symbolic matches or mismatches of concrete labels in the parse trees to compute the kernels. It is unclear at the first glance how to encode word embeddings into the tree kernels effectively so that word embeddings could help to improve the generalization performance of RE. One way is to use word embeddings to compute similarities between words and embed these similarity scores into the kernel functions, e.g., by resembling the method of Plank and Moschitti (2013) that exploited LSA (in the semantic syntactic tree kernel (SSTK), cf. §2.1). We explore various methods to apply word embeddings to generate the semantic representations for DA of RE and demonstrate that semantic representations are very effective to significantly improve the portability of the relation extractors based on the tree kernels, bringing us to the second question:

*(ii) Between the feature-based method in Nguyen and Grishman (2014) and the SSTK method in Plank and Moschitti (2013), which method is better for DA of RE, given the recent discovery of word embeddings for both methods?*

It is worth noting that besides the approach difference, these two works employ rather different resources and settings in their evaluation, making it impossible to directly compare their performance. In particular, while Plank and Moschitti (2013) only use the path-enclosed trees induced from the constituent parse trees as the representation for relation mentions, Nguyen and Grishman (2014) include a rich set of features extracted from multiple resources such as constituent trees, dependency trees, gazetteers, semantic resources in the representation. Besides, Plank and Moschitti (2013) consider the direction of relations in their evaluation (i.e, distinguishing between relation classes and their inverses) but Nguyen and Grishman (2014) disregard this relation direction. Finally, we note that although both studies evaluate their systems on the ACE 2005 dataset, they actually have different dataset partitions. In order to overcome this limitation, we conduct an evaluation in which the two methods are directed to use the same resources and settings, and are thus compared in a *compatible* manner to achieve an insight on their effectiveness for DA of RE. In fact, the problem of incompatible comparison is unfortunately very common in the RE literature (Wang, 2008; Plank and Moschitti, 2013) and we believe there is a need to tackle this increasing confusion in this line of research. Therefore, this is actually the first attempt to compare the two methods (tree kernel-based and feature-based) on the same settings. To ease the comparison for future work and circumvent the *Zigglebottom* pitfall (Pedersen, 2008), the entire setup and package is available.[1]

## 2 Relation Extraction Approaches

In the following, we introduce the two relation extraction systems further examined in this study.

### 2.1 Tree kernel-based Method

In the tree kernel-based method (Moschitti, 2006; Moschitti, 2008; Plank and Moschitti, 2013), a relation mention (the two entity mentions and the sentence containing them) is represented by the path-enclosed tree (PET), the smallest constituency-based subtree including the two target entity mentions (Zhang et al., 2006). The syntactic tree kernel (STK) is then defined to compute the similarity between two PET trees (where target entities are marked) by counting the common sub-trees, without enumerating the whole fragment space (Moschitti, 2006; Moschitti, 2008). STK is then applied in the support vector machines (SVMs) for RE. The major limitation of STK is its inability to match two trees that share the same substructure, but involve different though semantically related terminal nodes (words). This is caused by the hard matches between words, and consequently between sequences containing them. For instance, in the following example taken from Plank and Moschitti (2013), the two fragments "*governor from Texas*" and "*head of Mary-*

[1] https://bitbucket.org/nycphre/limo-re

*land*" would not match in STK although they have very similar syntactic structures and basically convey the same relationship.

Plank and Moschitti (2013) propose to resolve this issue for STK using the semantic syntactic tree kernel (SSTK) (Bloehdorn and Moschitti, 2007) and apply it to the domain adaptation problem of RE. The two following techniques are utilized to activate the SSTK: (i) replace the part-of-speech nodes in the PET trees by the new ones labeled by the word clusters of the corresponding terminals (words); (ii) replace the binary similarity scores between words (i.e, either 1 or 0) by the similarities induced from the latent semantic analysis (LSA) of large corpus. The former generalizes the part-of-speech similarity to the semantic similarity on word clusters; the latter, on the other hand, allows soft matches between words that have the same latent semantic but differ in symbolic representation. Both techniques emphasize the invariants of word semantics in different domains, thus being helpful to alleviate the vocabulary difference across domains.

### 2.2 Feature-based Method

In the feature-based method (Zhou et al., 2005; Sun et al., 2011; Nguyen and Grishman, 2014), relation mentions are first transformed into rich feature vectors that capture various characteristics of the relation mentions (i.e, lexicon, syntax, semantics etc). The resulting vectors are then fed into the statistical classifiers such as Maximum Entropy (MaxEnt) to perform classification for RE.

The main reason for the performance loss of the feature-based systems on new domains is the behavioral changes of the features when domains shift. Some features might be very informative in the source domain but become less relevant in the target domains. For instance, some words, that are very indicative in the source domain might not appear in the target domains (lexical sparsity). Consequently, the models putting high weights on such words (features) in the source domain will fail to perform well on the target domains. Nguyen and Grishman (2014) address this problem for the feature-based method in DA of RE by introducing word embeddings as additional features. The rationale is based on the fact that word embeddings are low dimensional and real valued vectors, capturing latent syntactic and semantic properties of words (Bengio et al., 2003; Mnih and Hinton, 2008; Turian et al., 2010). The embeddings of symbolically different words are often close to each other if they have similar semantic and syntactic functions. This again helps to mitigate the lexical sparsity or the vocabulary difference between the domains and has proven helpful for, amongst others, the feature-based method in DA of RE.

### 2.3 Tree Kernel-based vs Feature-based

The feature-based method explicitly encapsulates the linguistic intuition and domain expertise for RE into the features, while the tree kernel-based method avoids the complicated feature engineering and implicitly encode the features into the computation of the tree kernels. Which method is better for DA of RE?

In order to ensure the two methods (Plank and Moschitti, 2013; Nguyen and Grishman, 2014) are compared compatibly on the same resources, we make sure the two systems have access to the same amount of information. Thus, we follow Plank and Moschitti (2013) and use the PET trees (beside word clusters and word embeddings) as the only resource the two methods can exploit.

For the feature-based method, we utilize all the features extractable from the PET trees that are standardly used in the state-of-the-art feature-based systems for DA of RE (Nguyen and Grishman, 2014). Specifically, the feature set employed in this paper (denoted by FET) includes: the lexical features, i.e., the context words, the head words, the bigrams, the number of words, the lexical path, the order of mention (Zhou et al., 2005; Sun et al., 2011); and the syntactic features, i.e., the path connecting the two mentions in PET and the unigrams, bigrams, trigrams along this path (Zhou et al., 2005; Jiang and Zhai, 2007a).

*Hypothesis*: Assuming identical settings and resources, we hypothesize that the tree kernel-based method is better than the feature-based method for DA of RE. This is motivated because of at least two reasons: (i) the tree kernel-based method implicitly encodes a more comprehensive feature set (involving all the sub-trees in the PETs), thus potentially captures more domain-independent features to be useful for DA of RE; (ii) the tree kernel-based method avoids the inclusion of fine-tuned and domain-specific features originated from the excessive feature engineering (i.e., hand-designing feature sets based on the

linguistic intuition for specific domains) of the feature-based method.

## 3 Word Embeddings & Tree Kernels

In this section, we first give the intuition that guides us in designing the proposed methods. In particular, one limitation of the syntactic semantic tree kernel presented in Plank and Moschitti (2013) (§2.1) is that semantics is highly tied to syntax (the PET trees) in the kernel computation, limiting the generalization capacity of semantics to the extent of syntactic matches. If two relation mentions have different syntactic structures, the two relation mentions will not match, although they share the same semantic representation and express the same relation class. For instance, the two fragments "*Tom is the CEO of the company*" and "*the company, headed by Tom*" express the same relationship between "*Tom*" and "*company*" based on the semantics of their context words, but cannot be matched in SSTK as their syntactic structures are different. In such a case, it is desirable to have a representation of relation mentions that is grounded on the semantics of the context words and reflects the latent semantics of the whole relation mentions. This representation is expected to be general enough to be effective on different domains. Once the semantic representation of relation mentions is established, we can use it in conjunction with the traditional tree kernels to extend their coverage. The benefit is mutual as both semantics and syntax help to generalize relation mentions to improve the recall, but also constrain each other to support precision. This is the basic idea of our approach, which we compare to the previous methods.

### 3.1 Methods

We propose to utilize word embeddings of the context words as the principal components to obtain semantic representations for relation mentions in the tree kernel-based methods. Besides more traditional approaches to exploit word embeddings, we investigate representations that go beyond the word level and use compositionality embeddings for domain adaptation for the first time.

In general, suppose we are able to acquire an additional real-valued vector $V_i$ from word embeddings to semantically represent a relation mention $R_i$ (along with the PET tree $T_i$), leading to the new representation of $R_i = (T_i, V_i)$. The new kernel function in this case is then defined by:

$$K_{new}(R_i, R_j) = (1 - \alpha)\text{SSTK}(T_i, T_j) + \alpha K_{vec}(V_i, V_j)$$

where $K_{vec}(V_i, V_j)$ is some standard vector kernel like the polynomial kernels. $\alpha$ is a trade-off parameter and indicates whether the system attributes more weight to the traditional SSTK or the new semantic kernel $K_{vec}$.

In this work, we consider the following methods to obtain the semantic representation $V_i$ from the word embeddings of the context words of $R_i$ (assuming $d$ is the dimensionality of the word embeddings):

**HEAD**: $V_i$ = the concatenation of the word embeddings of the two entity mention heads of $R_i$. This representation is inherited from Nguyen and Grishman (2014) that only examine embeddings at the word level separately for the feature-based method without considering the compositionality embeddings of relation mentions. The dimensionality of HEAD is $2d$.

According to the principle of compositionality (Werning et al., 2006; Baroni and Zamparelli, 2010; Paperno et al., 2014), the meaning of a complex expression is determined by the meanings of its components and the rules to combine them. We study the following two compositionality embeddings for relation mentions that can be generated from the embeddings of the context words:

**PHRASE**: $V_i$ = the mean of the embeddings of the words contained in the PET tree $T_i$ of $R_i$. Although this composition is simple, it is in fact competitive to the more complicated methods based on recursive neural networks (Socher et al., 2012b; Blacoe and Lapata, 2012; Sterckx et al., 2014) on representing phrase semantics.

**TREE**: This is motivated by the training of recursive neural networks (Socher et al., 2012a) for semantic compositionality and attempts to aggregate the context words embeddings syntactically. In particular, we compute an embedding for every node in the PET tree in a bottom-up manner. The embeddings of the leaves are the embeddings of the words associated with them while the embeddings of the internal nodes are the means of the embeddings of their children nodes. We use the embeddings of the root of the PET tree to represent the relation mention in this case. Both PHRASE and TREE have $d$ dimensions.

It is also interesting to examine combinations of these three representations (cf., Table 1).

**SIM**: Finally, for completeness, we experiment with a more obvious way to introduce word embeddings into tree kernels, resembling more closely the approach of Plank and Moschitti (2013). In particularly, the SIM method simply replaces the similarity scores between word pairs obtained from LSA by the cosine similarities between the word embeddings to be used in the SSTK kernel.

## 4 Experiments

### 4.1 Dataset, Resources and Parameters

We use the word clusters trained by Plank and Moschitti (2013) on the ukWaC corpus (Baroni et al., 2009) with 2 billion words, and the C&W word embeddings from Turian el al. (2010)[2] with 50 dimensions following Nguyen and Grishman (2014). In order to make the comparisons compatible, we introduce word embeddings into the tree kernel by extending the package provided by Plank and Moschitti (2013), which uses the Charniak parser to obtain the constituent trees, the SVM-light-TK for the SSTK kernel in SVM, the directional relation classes, etc. We utilize the default vector kernel in the SVM-light-TK package (`d=3`). For the feature-based method, we apply the MaxEnt classifier in the MALLET[3] package with the L2 regularizer on the hierarchical architecture for relation extraction as in Nguyen and Grishman (2014).

Following prior work, we evaluate the systems on the ACE 2005 dataset which involves 6 domains: broadcast news (bn), newswire (nw), broadcast conversation (bc), telephone conversation (cts), weblogs (wl) and usenet (un). The union of **bn** and **nw** (**news**) is used as the source domain while **bc**, **cts** and **wl** play the role of the target domains. We take half of bc as the only target development set, and use the remaining data and domains for testing. The dataset partition is exactly the same as in Plank and Moschitti (2013). As described in their paper, the target domains quite differ from the source domain in the relation distributions and vocabulary.

### 4.2 Word Embeddings for Tree Kernel

We investigate the effectiveness of different semantic representations (§3.1) in tree kernels by taking the PET tree as the baseline[4], and evaluate the performance of the representations when combined with the baseline on the bc development set.

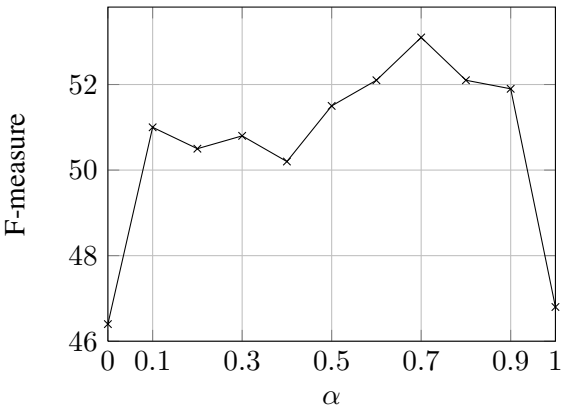


Figure 1: $\alpha$ vs F-measure on PET+HEAD+PHRASE

| Method | P | R | F1 |
|---|---|---|---|
| PET (Plank and Moschitti, 2013) | 52.2 | 41.7 | 46.4 |
| PET+SIM | 39.4 | 37.2 | 38.3 |
| PET+HEAD | 60.4 | 44.9 | 51.5 |
| PET+PHRASE | 58.4 | 40.7 | 48.0 |
| PET+TREE | 59.8 | 42.2 | 49.5 |
| **PET+HEAD+PHRASE** | 63.2 | 46.2 | **53.4** |
| PET+HEAD+TREE | 61.0 | 45.7 | 52.3 |
| PET+PHRASE+TREE | 59.2 | 42.4 | 49.4 |
| PET+HEAD+PHRASE+TREE | 60.8 | 45.2 | 51.9 |

Table 1: Performance on the bc dev set for PET. Best combination (HEAD+PHRASE) is denoted WED in Table 2

Table 1 shows the results. The main conclusions include:

(i) The substitution of LSA similarity scores with the word embedding cosine similarities (SIM) does not help to improve the performance of the tree kernel method.

(ii) When employed independently, both the word level embeddings (HEAD) and the compositionality embeddings (PHRASE, TREE) are effective for the tree kernel-based method on DA for RE, showing a slight advantage for HEAD.

(iii) Thus, the compositionality embeddings PHRASE and TREE seem to capture different information with respect to the word level embeddings HEAD. We expect the combination of HEAD with either PHRASE or TREE to further improve performance. This is the case when adding one of them at a time. PHRASE and TREE seem to capture similar information, combining all (last row in Table 1) is not the overall best system. The best performance is achieved when the HEAD and PHRASE embeddings are utilized at

[2] http://metaoptimize.com/projects/wordreprs/

[3] http://mallet.cs.umass.edu/

[4] By using their system we obtained the same results.

| # | System: | **nw+bn** (in-dom.) P: | R: | F1: | **bc** P: | R: | F1: | **cts** P: | R: | F1: | **wl** P: | R: | F1: |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | PET (Plank and Moschitti, 2013) | 50.6 | 42.1 | 46.0 | 51.2 | 40.6 | 45.3 | 51.0 | 37.8 | 43.4 | 35.4 | 32.8 | 34.0 |
| 2 | PET+WED | 55.8 | 48.7 | 52.0 | 57.3 | 45.7 | 50.8 | 54.0 | 38.1 | 44.7 | 40.1 | 36.5 | 38.2 |
| 3 | PET_WC | 55.4 | 44.6 | 49.4 | 54.3 | 41.4 | 47.0 | 55.9 | 37.1 | 44.6 | 40.0 | 32.7 | 36.0 |
| 4 | PET_WC+WED | 56.3 | 48.2 | 51.9 | 57.0 | 44.3 | 49.8 | 56.1 | 38.1 | 45.4 | 40.7 | 36.1 | 38.2 |
| 5 | PET_LSA | 52.3 | 44.1 | 47.9 | 51.4 | 41.7 | 46.0 | 49.7 | 36.5 | 42.1 | 38.1 | 36.5 | 37.3 |
| 6 | PET_LSA+WED | 55.2 | 48.5 | 51.6 | 58.8 | 45.8 | 51.5 | 54.1 | 38.1 | 44.7 | 40.9 | 38.5 | 39.6 |
| 7 | PET+PET_WC | 55.0 | 46.5 | 50.4 | 54.4 | 43.4 | 48.3 | 54.1 | 38.1 | 44.7 | 38.4 | 34.5 | 36.3 |
| 8 | PET+PET_WC+WED | 56.3 | 50.3 | 53.1 | 57.5 | 46.6 | 51.5 | 55.6 | 39.8 | **46.4** | 41.5 | 37.9 | 39.6 |
| 9 | PET+PET_LSA | 52.7 | 46.6 | 49.5 | 53.9 | 45.2 | 49.2 | 49.9 | 37.6 | 42.9 | 37.9 | 38.3 | 38.1 |
| 10 | PET+PET_LSA+WED | 55.5 | 49.9 | 52.6 | 56.8 | 45.8 | 50.8 | 52.5 | 38.6 | 44.5 | 41.6 | 39.3 | **40.5** |
| 11 | PET+PET_WC+PET_LSA | 55.1 | 45.9 | 50.1 | 55.3 | 43.1 | 48.5 | 53.1 | 37.0 | 43.6 | 39.9 | 35.8 | 37.8 |
| 12 | PET+PET_WC+PET_LSA+WED | 55.0 | 48.8 | 51.7 | 58.5 | 47.3 | **52.3** | 52.6 | 38.8 | 44.7 | 42.3 | 38.9 | **40.5** |

Table 2: In-domain (first column) and out-of-domain performance (columns two to four) on ACE 2005. Systems of the rows not in gray come from Plank and Moschitti (2013) (the baselines). WED means HEAD+PHRASE.

the same time, reaching an F1 of 53.4% (compared to 46.4% of the baseline) on the development set.

The results in Table 1 are obtained using the trade-off parameter $\alpha = 0.7$. Figure 1 additionally shows the variation of the performance with changing $\alpha$ (for the best system on dev, i.e., for the representation PET+HEAD+PHRASE). As we can see, the performance for $\alpha > 0.5$ is in general better, suggesting a preference for the semantic representation over the syntactic representation in DA for RE. The performance reaches its peak when the suitable amounts of semantics and syntax are combined (i.e, $\alpha = 0.7$).

In the following experiments, we use the embedding combination (HEAD+PHRASE) with $\alpha = 0.7$ for the tree kernels, denoted WED.

### 4.3 Domain Adaptation Experiments

In this section, we examine the semantic representation for DA of RE in the tree kernel-based method. In particular, we take the systems using the PET trees, word clusters and LSA in Plank and Moschitti (2013) as the baselines and augment them with the embeddings WED = HEAD+PHRASE. We report the performance of these augmented systems in Table 2 for the two scenarios: (i) in-domain: both training and testing are performed on the source domain via 5-fold cross validation and (ii) out-of-domain: models are trained on the source domain but evaluated on the three target domains. To summarize, we find:

First, word embeddings seem to subsume word clusters in the tree kernel-based method (comparing rows 2 and 4, and except domain cts) while word embeddings and LSA actually encode different information (comparing rows 2 and 6 for the out-of-domain experiments) and their combination would be helpful for DA of RE.

Second, regarding composite kernels, given word embeddings, the addition of the baseline kernel (PET) is in general useful for the augmented kernels PET_WC and PET_LSA (comparing rows 4 and 8, rows 6 and 10) although it is less pronounced for PET_LSA.

Third and most importantly, for all the systems in Plank and Moschitti (2013) (the baselines) and for all the target domains, whether word clusters and LSA are utilized or not, we consistently witness the performance improvement of the baselines when combined with word embedding (comparing systems X and X+WED where X is some baseline system). The best out-of-domain performance is achieved when word embeddings are employed in conjunction with the composite kernels (PET+PET_WC+PET_LSA for the target domains bc and wl, and PET+PET_WC for the target domain cts). To be more concrete, the best system with word embeddings (row 12 in Table 2) significantly outperforms the best system in Plank and Moschitti (2013) with $p < 0.05$, an improvement of 3.7%, 1.1% and 2.7% on the target domains bc, cts and wl respectively, demonstrating the benefit of word embeddings for DA of RE in the tree kernel-based method.

### 4.4 Tree Kernel-based vs Feature-based DA of RE

This section aims to compare the tree kernel-based method in Plank and Moschitti (2013) and the feature-based method in Nguyen and Grishman (2014) for DA of RE on the same settings (i.e, same dataset partition, the same pre-processing

| System: | **nw+bn** (in-dom.) P: | R: | F1: | **bc** P: | R: | F1: | **cts** P: | R: | F1: | **wl** P: | R: | F1: |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Tree kernel-based:** | | | | | | | | | | | | |
| PET+PET_WC+HEAD+PHRASE | 56.3 | 50.3 | **53.1** | 57.5 | 46.6 | **51.5** | 55.6 | 39.8 | **46.4** | 41.5 | 37.9 | **39.6** |
| **Feature-based:** | | | | | | | | | | | | |
| FET+WC+HEAD | 44.5 | 51.0 | 47.5 | 46.5 | 49.3 | 47.8 | 44.5 | 40.0 | 42.1 | 35.4 | 39.5 | 37.3 |
| FET+WC+TREE | 44.4 | 50.2 | 47.1 | 46.4 | 48.7 | 47.6 | 43.7 | 40.3 | 41.9 | 32.7 | 36.7 | 34.6 |
| FET+WC+HEAD+PHRASE | 44.9 | 51.6 | 48.0 | 46.0 | 49.1 | 47.5 | 45.2 | 41.5 | 43.3 | 34.7 | 39.2 | 36.8 |
| FET+WC+HEAD+TREE | 45.1 | 51.0 | 47.8 | 46.9 | 48.4 | 47.6 | 43.8 | 39.5 | 41.5 | 34.7 | 38.8 | 36.6 |

Table 3: Tree kernel-based in Plank and Moschitti (2013) vs feature-based in Nguyen and Grishman (2014). All the comparisons between the tree kernel-based method and the feature-based method in this table are significant with $p < 0.05$.

procedure, the same model of directional relation classes, the same PET trees for tree kernels and feature extraction, the same word clusters and the same word embeddings). We first evaluate the feature-based system with different combinations of embeddings (i.e, HEAD, PHRASE and TREE) on the bc development set. Based on the evaluation results, we then discuss the effect of the semantic representations on the feature-based system and the tree kernel-based system, and then compare the performance of the two methods when they are augmented with their best corresponding embedding combinations.

| System | P | R | F1 |
|---|---|---|---|
| B | 51.2 | 49.4 | 50.3 |
| B+HEAD | 55.8 | 52.4 | **54.0** |
| B+PHRASE | 50.7 | 46.2 | 48.4 |
| B+TREE | 53.6 | 51.1 | **52.3** |
| B+HEAD+PHRASE | 53.2 | 50.1 | 51.6 |
| B+HEAD+TREE | 54.9 | 51.4 | **53.1** |
| B+PHRASE+TREE | 50.7 | 48.4 | 49.5 |
| B+HEAD+PHRASE+TREE | 52.7 | 49.4 | 51.0 |

Table 4: Performance of the feature-based method (dev).

Table 4 presents the evaluation results on the bc development for the feature-based system where B is the baseline feature set consisting of FET and word clusters (WC) (Nguyen and Grishman, 2014).

**The Role of Semantic Representations** Considering Table 4 for the feature-based method and Table 1 for the tree kernel-based method, we see that when combined with the HEAD embeddings, the compositionality embedding TREE is more effective for the feature-based method, in contrast to the tree kernel-based method, where the PHRASE embeddings are better. This can be partly explained by the fact that the tree kernel-based method emphasizes the syntactic structure of the relation mentions, while the feature-based method exploits the sequential structure more. Consequently, the syntactic semantics of TREE are more helpful for the feature-based method, whereas the sequential semantics of PHRASE are more useful for the tree kernel-based method.

**Performance Comparison** The three best embedding combinations for the feature-based system in Table 4 are (listed by performance order): (HEAD), (HEAD+TREE) and (TREE), where (HEAD) is also the best word level method employed in Nguyen and Grishman (2014). In order to enable a fairer and clearer evaluation, when doing comparison, we use both the three best embedding combinations in the feature-based method and the best embedding combination (HEAD+PHRASE) in the tree kernel-based method. In the tree kernel-based method, we do not employ the LSA information as it comes in the form of similarity scores between pairs of words, and it is not clear how to encode this information into the feature-based method effectively. Finally, we utilize the composite kernel for its demonstrated effectiveness in Section 4.3.

The most important observation from the experimental results (shown in Table 3) is that over all the target domains, the tree kernel-based system is significantly better than the feature-based systems with $p < 0.05$ (assuming the same resources and settings mentioned above). In fact, there are large margins between the tree kernel-based and the feature-based methods in this case (i.e, about 3.7% for bc, 3.1% for cts and 2.3% for wl), clearly confirming the hypothesis about the advantage of the tree kernel-based method over the feature-based method on DA for RE in Section 2.3.

## 5 Analysis

This section analyzes the output of the systems to gain more insights into their operation.

**Word Embeddings for the Tree-kernel based Method** We focus on the comparison of the best model in Plank and Moschitti (2013) (row 11 in Table 2) (called P) with the same model but augmented with the embedding WED (row 12 in Tabel 2) (called P+WED). One of the most interesting insights is that the embedding WED helps to semantically generalize the phrases connecting the two target entity mentions beyond the syntactic constraints. For instance, model P fails to discover the relation between "*Chuck Hagel*" and "*Vietnam*" in the sentence (of the target domain bc): "*Sergeant Chuck Hagel was seriously wounded twice in Vietnam.*" (i.e, it returns the NONE relation as the prediction) as the substructure associated with "*seriously wounded twice*" does not appear with any relation in the source domain. Model P+WED, on the other hand, correctly predicts the PHYS (Located) relation between the two entities as the PHRASE embedding of "*Chuck Hagel was seriously wounded twice in Vietnam.*" (phrase X1) is very close to the embedding of the source domain phrase: "*Stewart faces up to 30 years in prison*" (phrase X2) (annotated with the PHYS relation between "*Stewart*" and "*prison*").

In fact, X2 is only the 9th closest phrase in the source domain of X1. The closest phrase of X1 in the source domain is X3: the phrase between "*Iraqi soldiers*" and "*herself*" in the sentence "*The Washington Post is reporting she shot several* ***Iraqi soldiers before she was captured and she was shot herself****, too.*". However, as the syntactical structure of X1 is more similar to X2's, and is remarkably different from X3 as well as the other closest phrases (ranked from 2nd to 8th), the new kernel function $K_{new}$ would still prefer X2 due to its trade-off between syntax and semantics.

**Tree Kernel-based vs Feature-based** From the analysis of the systems in Table 3, we find that, among others, the tree kernel-based method improves the precision significantly via the semantic and syntactic refinement it maintains. Let us consider the following phrase of the target domain bc: "*troops have dislodged stubborn Iraqi soldiers*" (called Y1). The feature-based systems in Table 3 incorrectly predict the ORG-AFF relation (Employment or Membership) between "*Iraqi soldiers*" and "*troops*". This is mainly due to the high weights of the features linking the words "*troop*" and "*soldiers*" with the relation type ORG-AFF in the feature-based models, which is, in turn, originated from the high correlation of these words and the relation type in the training data of the source domain (domain bias). The tree kernel-based model in Table 3 successfully recognizes the NONE relation in this case. A closer examination shows that the phrase with the closest embedding to Y1 in the source domain is Y2: "*Iraqi soldiers abandoned their posts*",[5] which is annotated with the NONE relation between "*Iraqi soldiers*" and "*their posts*". As the syntactic structure of Y2 is also very similar to Y1, it is not surprising that Y1 is closest to Y2 in the new kernel function, consequently helping the tree kernel-based method work correctly in this case.

## 6 Related work

Word embeddings are only applied to RE recently. Socher et al. (2012b) use word embeddings as input for matrix-vector recursive neural networks in relation classification while Zeng et al. (2014), and Nguyen and Grishman (2015) employ word embeddings in the framework of convolutional neural networks for relation classification and extraction, respectively. Sterckx et al. (2014) utilize word embeddings to reduce noise of training data in distant supervision. Kuksa et al. (2010) present a string kernel for bio-relation extraction with word embeddings, and Yu et al. (2014; 2015) study the factor-based compositional embedding models. However, none of this work examines word embeddings for tree kernels as well as domain adaptation as we do.

Regarding DA, in the unsupervised DA setting, Huang and Yates (2010) attempt to learn multi-dimensional feature representations while Blitzer et al. (2006) introduce structural correspondence learning. Daumé (2007) proposes an easy adaptation framework (EA) while Xiao and Guo (2013) present a log-bilinear language adaptation technique in the supervised DA setting. Unfortunately, all of this work assumes some prior (in the form of either labeled or unlabeled data) on the target domains for the sequential labeling tasks, in contrast to our single-system unsupervised DA setting for relation extraction. An alternative method that is also popular to DA is instance weighting (Jiang and Zhai, 2007b). However, as shown by Plank and Moschitti (2013), instance weighting is not

[5] The full sentence is: "*After today's air strikes, Iraqi soldiers abandoned their posts and surrendered to Kurdish fighters.*".

very useful for DA of RE.

## 7 Conclusion

In order to improve the generalization of relation extractors, we propose to augment the semantic syntactic tree kernels with the semantic representation of relation mentions, generated from the word embeddings of the context words. The method demonstrates strong promise for the DA of RE, i.e, it significantly improves the best system of Plank and Moschitti (2013) (up to 7% relative improvement). Moreover, we perform a compatible comparison between the tree kernel-based method and the feature-based method on the same settings and resources, which suggests that the tree kernel-based method (Plank and Moschitti, 2013) is better than the feature-based method (Nguyen and Grishman, 2014) for DA of RE. An error analysis is conducted to get a deeper comprehension of the systems. Our future plan is to investigate other syntactic and semantic structures (such as dependency trees, abstract meaning representation etc) for DA of RE, as well as continue the comparison between the kernel-based method and the feature-based method when they are allowed to exploit more resources.

## References


Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *EMNLP*.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. In *Language Resources and Evaluation*, pages 209–226.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. In *Journal of Machine Learning Research 3*, pages 1137–1155.

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *EMNLP*.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *EMNLP*.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*.

Stephan Bloehdorn and Alessandro Moschitti. 2007. Exploiting Structure and Semantics for Expressive Text Kernels. In *CIKM*.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. In *Computational Linguistics*, pages 467–479.

Razvan C. Bunescu and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *EMNLP*.

Razvan C. Bunescu. 2007. Learning to extract relations from the web using minimal supervision. In *ACL*.

Yee Seng Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. In *COLING*.

Hal Daume. 2007. Frustratingly easy domain adaptation. In *ACL*.

Ralph Grishman, David Westbrook, and Adam Meyers. 2005. Nyu's english ace 2005 system description. In *The ACE 2005 Evaluation Workshop*.

Fei Huang and Alexander Yates. 2010. Exploring representation-learning approaches to domain adaptation. In *The ACL Workshop on Domain Adaptation for Natural Language Processing (DANLP)*.

Jing Jiang and ChengXiang Zhai. 2007a. A systematic exploration of the feature space for relation extraction. In *NAACL-HLT*.

Jing Jiang and ChengXiang Zhai. 2007b. Instance weighting for domain adaptation in nlp. In *ACL*.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *ACL*.

Pavel Kuksa, Yanjun Qi, Bing Bai, Ronan Collobert, Jason Weston, Vladimir Pavlovic, and Xia Ning. 2010. Semi-supervised abstraction-augmented string kernel for multi-level bio-relation extraction. In *ECML PKDD*.

Andriy Mnih and Geoffrey Hinton. 2008. A scalable hierarchical distributed language model. In *NIPS*.

Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *ECML*.

Alessandro Moschitti. 2008. Kernel methods, syntax and semantics for relational text categorization. In *CIKM*.

Thien Huu Nguyen and Ralph Grishman. 2014. Employing word representations and regularization for domain adaptation of relation extraction. In *ACL*.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *The NAACL Workshop on Vector Space Modeling for NLP (VSM)*.

T. Truc-Vien Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *EMNLP*.

Luan Minh Nguyen, W. Ivor Tsang, A. Kian Ming Chai, and Leong Hai Chieu. 2014. Robust domain adaptation for relation extraction via clustering consistency. In *ACL*.

Denis Paperno, The Nghia Pham, and Marco Baroni. 2014. A practical and linguistically-motivated approach to compositional distributional semantics. In *ACL*.

Ted Pedersen. 2008. Empiricism is not a matter of faith. In *Computational Linguistics 3*, pages 465–470.

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 shared task on parsing the web. In *The First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

Barbara Plank and Alessandro Moschitti. 2013. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *ACL*.

Longhua Qian, Guodong Zhou, Fang Kong, Qiaoming Zhu, and Peide Qian. 2008. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In *COLING*.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2012a. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP-CoNLL*.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012b. Semantic compositionality through recursive matrix-vector spaces. In *EMNLP*.

Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. 2014. Using active learning and semantic clustering for noise reduction in distant supervision. In *AKBC*.

Ang Sun, Ralph Grishman, and Satoshi Sekine. 2011. Semi-supervised relation extraction with large-scale word clustering. In *ACL*.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *ACL*.

Mengqiu Wang. 2008. A re-examination of dependency path kernels for relation extraction. In *IJCNLP*.

Markus Werning, Edouard Machery, and Gerhard Schurz. 2006. Compositionality of meaning and content: Foundational issues (linguistics & philosophy). In *Linguistics & philosophy*.

Min Xiao and Yuhong Guo. 2013. Domain adaptation for sequence labeling tasks with a probabilistic language adaptation model. In *ICML*.

Mo Yu, Matthew Gormley, and Mark Dredze. 2014. Factor-based compositional embedding models. In *The NIPS workshop on Learning Semantics*.

Mo Yu, Matthew Gormley, and Mark Dredze. 2015. Combining word embeddings and feature embeddings for fine-grained relation extraction. In *NAACL*.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. In *Journal of Machine Learning Research 3*, pages 1083–1106.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *COLING*.

Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. 2006. A composite kernel to extract relations between entities with both flat and structured features. In *COLING-ACL*.

Shubin Zhao and Ralph Grishman. 2005. Extracting relations with integrated information using kernel methods. In *ACL*.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *ACL*.