Embedding Semantic Similarity in Tree Kernels for Domain Adaptation of Relation Extraction

Barbara Plank*
Center for Language Technology
University of Copenhagen, Denmark
bplank@gmail.com

Alessandro Moschitti
QCRI - Qatar Foundation &
DISI - University of Trento, Italy
amoschitti@qf.org.qa

Abstract

Relation Extraction (RE) is the task of extracting semantic relationships between entities in text. Recent studies on relation extraction are mostly supervised. The clear drawback of supervised methods is the need of training data: labeled data is expensive to obtain, and there is often a mismatch between the training data and the data the system will be applied to. This is the problem of *domain adaptation*. In this paper, we propose to combine (i) term generalization approaches such as word clustering and latent semantic analysis (LSA) and (ii) structured kernels to improve the adaptability of relation extractors to new text genres/domains. The empirical evaluation on ACE 2005 domains shows that a suitable combination of syntax and lexical generalization is very promising for domain adaptation.

1 Introduction

Relation extraction is the task of extracting semantic relationships between entities in text, e.g. to detect an employment relationship between the person *Larry Page* and the company *Google* in the following text snippet: *Google CEO Larry Page holds a press announcement at its headquarters in New York on May 21, 2012*. Recent studies on relation extraction have shown that supervised approaches based on either feature or kernel methods achieve state-of-the-art accuracy (Zelenko et al., 2002; Culotta and Sorensen, 2004; Zhang et al., 2005; Zhou et al., 2005; Zhang et al., 2006; Bunescu, 2007; Nguyen et al., 2009; Chan and Roth, 2010; Sun et al., 2011). However, the clear drawback of supervised methods is the need of training data, which can slow down the delivery of commercial applications in new domains: labeled data is expensive to obtain, and there is often a mismatch between the training data and the data the system will be applied to. Approaches that can cope with domain changes are essential. This is the problem of domain adaptation (DA) or transfer learning (TL). Technically, domain adaptation addresses the problem of learning when the assumption of independent and identically distributed (i.i.d.) samples is violated. Domain adaptation has been studied extensively during the last couple of years for various NLP tasks, e.g. two shared tasks have been organized on domain adaptation for dependency parsing (Nivre et al., 2007; Petrov and McDonald, 2012). Results were mixed, thus it is still a very active research area.

However, to the best of our knowledge, there is almost no work on adapting relation extraction (RE) systems to new domains.[1] There are some prior studies on the related tasks of *multi-task transfer learning* (Xu et al., 2008; Jiang, 2009) and *distant supervision* (Mintz et al., 2009), which are clearly related but different: the former is the problem of how to transfer knowledge from old to new relation types, while distant supervision tries to learn new relations from unlabeled text by exploiting weak-supervision in the form of a knowledge resource (e.g. Freebase). We assume the same relation types but a shift in the underlying

* The first author was affiliated with the Department of Computer Science and Information Engineering of the University of Trento (Povo, Italy) during the design of the models, experiments and writing of the paper.

[1] Besides an unpublished manuscript of a student project, but it is not clear what data was used. http://tinyurl.com/bn2hdwk

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 1498–1507,
Sofia, Bulgaria, August 4-9 2013.

data distribution. Weak supervision is a promising approach to improve a relation extraction system, especially to increase its coverage in terms of types of relations covered. In this paper we examine the related issue of changes in the underlying data distribution, while keeping the relations fixed. Even a weakly supervised system is expected to perform well when applied to any kind of text (other domain/genre), thus ideally, we believe that combining domain adaptation with weak supervision is the way to go in the future. This study is a first step towards this.

We focus on *unsupervised* domain adaptation, i.e. no labeled target data. Moreover, we consider a particular domain adaptation setting: *single-system DA*, i.e. learning a single system able to cope with different but related domains. Most studies on DA so far have focused on building a specialized system for every specific target domain, e.g. Blitzer et al. (2006). In contrast, the goal here is to build a single system that can robustly handle several domains, which is in line with the setup of the recent shared task on parsing the web (Petrov and McDonald, 2012). Participants were asked to build a single system that can robustly parse all domains (reviews, weblogs, answers, emails, newsgroups), rather than to build several domain-specific systems. We consider this as a shift in what was considered domain adaptation in the past (adapt from source to a specific target) and what can be considered a somewhat different recent view of DA, that became widespread since 2011/2012. The latter assumes that the target domain(s) is/are not really known in advance. In this setup, the domain adaptation problem boils down to finding a more robust system (Søgaard and Johannsen, 2012), i.e. one wants to build a system that can robustly handle any kind of data.

We propose to combine (i) term generalization approaches and (ii) structured kernels to improve the performance of a relation extractor on new domains. Previous studies have shown that lexical and syntactic features are both very important (Zhang et al., 2006). We combine structural features with lexical information generalized by clusters or similarity. Given the complexity of feature engineering, we exploit kernel methods (Shawe-Taylor and Cristianini, 2004). We encode word clusters or similarity in tree kernels, which, in turn, produce spaces of tree fragments. For example, "president", "vice-president" and "Texas", "US", are terms indicating an employment relation between a person and a location. Rather than only matching the surface string of words, lexical similarity enables *soft* matches between similar words in convolution tree kernels. In the empirical evaluation on Automatic Content Extraction (ACE) data, we evaluate the impact of convolution tree kernels embedding lexical semantic similarities. The latter is derived in two ways with: (a) Brown word clustering (Brown et al., 1992); and (b) Latent Semantic Analysis (LSA). We first show that our system aligns well with the state of the art on the ACE 2004 benchmark. Then, we test our RE system on the ACE 2005 data, which exploits kernels, structures and similarities for domain adaptation. The results show that combining the huge space of tree fragments generalized at the lexical level provides an effective model for adapting RE systems to new domains.

2 Semantic Syntactic Tree Kernels

In kernel-based methods, both learning and classification only depend on the inner product between instances. Kernel functions can be efficiently and implicitly computed by exploiting the dual formulation: $\sum_{i=1..l} y_i \alpha_i \phi(o_i)\phi(o) + b = 0$, where o_i and o are two objects, ϕ is a mapping from an object to a feature vector $\vec{x_i}$ and $\phi(o_i)\phi(o) = K(o_i, o)$ is a kernel function implicitly defining such a mapping. In case of structural kernels, K determines the shape of the substructures describing the objects. Commonly used kernels in NLP are string kernels (Lodhi et al., 2002) and tree kernels (Moschitti, 2006; Moschitti, 2008).

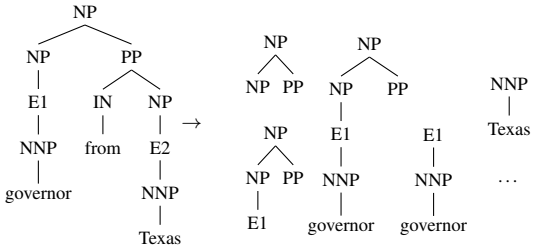

Figure 1: Syntactic tree kernel (STK).

Syntactic tree kernels (Collins and Duffy, 2001) compute the similarity between two trees T_1 and T_2 by counting common sub-trees (cf. Figure 1), without enumerating the whole fragment space. However, if two trees have similar substructures that employ different though related terminal nodes, they will not be matched. This is

clearly a limitation. For instance, the fragments corresponding to `governor from Texas` and `head of Maryland` are intuitively semantically related and should obtain a higher match when compared to `mother of them`.

Semantic syntactic tree kernels (Bloehdorn and Moschitti, 2007a; Bloehdorn and Moschitti, 2007b; Croce et al., 2011) provide one way to address this problem by introducing similarity σ that allows *soft matches* between words and, consequently, between fragments containing them. Let N_1 and N_2 be the set of nodes in T_1 and T_2, respectively. Moreover, let $I_i(n)$ be an indicator variable that is 1 if subtree i is rooted at n and 0 otherwise. The syntactic semantic convolution kernel TK_σ (Bloehdorn and Moschitti, 2007b) over $T1$ and $T2$ is computed as $TK_\sigma(T1, T2) = \sum_{n_1 \in N_1, n_2 \in N_2} \Delta_\sigma(n_1, n_2)$ where $\Delta_\sigma(n_1, n_2) = \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \sum_i I_i(n_1) I_i(n_2)$ is computed efficiently using the following recursive definition: i) If the nodes n_1 and n_2 are either different or have different number of children then $\Delta_\sigma(n_1, n_2) = 0$; else ii) If n_1 and n_2 are pre-terminals then $\Delta_\sigma(n_1, n_2) = \lambda \prod_{j=1}^{nc(n_1)} \Delta_\sigma(ch(n_1, j), ch(n_2, j))$, where σ measures the similarity between the corresponding children of n_1 and n_2; iii) If n_1 and n_2 have identical children: $\Delta_\sigma(n_1, n_2) = \lambda \prod_{j=1}^{nc(n_1)} (1 + \Delta_\sigma(ch(n_1, j)), ch(n_2, j))$; else $\Delta_\sigma(n_1, n_2) = 0$. TK_σ combines generalized lexical with structural information: it allows matching tree fragments that have the same syntactic structure but differ in their terminals. After introducing related work, we will discuss computational structures for RE and their extension with semantic similarity.

3 Related Work

Semantic syntactic tree kernels have been previously used for question classification (Bloehdorn and Moschitti, 2007a; Bloehdorn and Moschitti, 2007b; Croce et al., 2011). These kernels have not yet been studied for either domain adaptation or RE. Brown clusters were studied previously for feature-based approaches to RE (Sun et al., 2011; Chan and Roth, 2010), but they were not yet evaluated in kernels. Thus, we present a novel application of semantic syntactic tree kernels and Brown clusters for domain adaptation of tree-kernel based relation extraction.

Regarding domain adaptation, several methods have been proposed, ranging from instance weighting (Jiang and Zhai, 2007) to approaches that change the feature representation (Daumé III, 2007) or try to exploit pivot features to find a generalized shared representation between domains (Blitzer et al., 2006). The easy-adapt approach presented in Daumé III (2007) assumes the supervised adaptation setting and is thus not applicable here. Structural correspondence learning (Blitzer et al., 2006) exploits unlabeled data from both source and target domain to find correspondences among features from different domains. These correspondences are then integrated as new features in the labeled data of the source domain. The key to SCL is to exploit pivot features to automatically identify feature correspondences, and as such is applicable to feature-based approaches but not in our case since we do not assume availability of target domain data. Instead, we apply a similar idea where we exploit an entire unlabeled corpus as pivot, and compare our approach to *instance weighting* (Jiang and Zhai, 2007).

Instance weighting is a method for domain adaptation in which instance-dependent weights are assigned to the loss function that is minimized during the training process. Let $l(x, y, \theta)$ be some loss function. Then, as shown in Jiang and Zhai (2007), the loss function can be weighted by $\beta_i l(x, y, \theta)$, such that $\beta_i = \frac{P_t(x_i)}{P_s(x_i)}$, where P_s and P_t are the source and target distributions, respectively. Huang et al. (2007) present an application of instance weighting to support vector machines by minimizing the following re-weighted function: $\min_{\theta,\xi} \frac{1}{2}||\theta||^2 + C \sum_{i=1}^{m} \beta_i \xi_i$. Finding a good weight function is non-trivial (Jiang and Zhai, 2007) and several approximations have been evaluated in the past, e.g. Søgaard and Haulrich (2011) use a bigram-based text classifier to discriminate between domains. We will use a binary classifier trained on RE instance representations.

4 Computational Structures for RE

A common way to represent a constituency-based relation instance is the PET (path-enclosed-tree), the smallest subtree including the two target entities (Zhang et al., 2006). This is basically the former structure PAF[2] (predicate argument feature) defined in Moschitti (2004) for the extraction of predicate argument relations. The syntactic rep-

[2]It is the smallest subtree enclosing the predicate and one of its argument node.

resentation used by Zhang et al. (2006) (we will refer to it as PET Zhang) is the PET with enriched entity information: e.g. E1-NAM-PER, including entity type (PER, GPE, LOC, ORG) and mention type (NAM, NOM, PRO, PRE: name, nominal, pronominal or premodifier). An alternative kernel that does not use syntactic information is the Bag-of-Words (BOW) kernel, where a single root node is added above the terminals. Note that in this BOW kernel we actually mark target entities with E1/E2. Therefore, our BOW kernel can be considered an enriched BOW model. If we do not mark target entities, performance drops considerably, as discussed later.

As shown by Zhang et al. (2006), including gold-standard information on entity and mention type substantially improves relation extraction performance. We will use this gold information also in Section 6.1 to show that our system aligns well to the state of the art on the ACE 2004 benchmark. However, in a realistic setting this information is not available or noisy. In fact, as we discuss later, excluding gold entity information decreases system performance considerably. In the case of porting a system to new domains entity information will be unreliable or missing. Therefore, in our domain adaptation experiments on the ACE 2005 data (Section 6.3) we will not rely on this gold information but rather train a system using PET (target mentions only marked with E1/E2 and no gold entity label).[3]

4.1 Syntactic Semantic Structures

Combining syntax with semantics has a clear advantage: it generalizes lexical information encapsulated in syntactic parse trees, while at the same time syntax guides semantics in order to obtain an effective semantic similarity. In fact, lexical information is highly affected by data-sparseness, thus tree kernels combined with semantic information created from additional resources should provide a way to obtain a more robust system.

We exploit this idea here for domain adaptation (DA): if words are generalized by semantic similarity LS, then in a hypothetical world changing LS such that it reflects the target domain would allow the system to perform better in the target domain. The question remains how to establish a link between the semantic similarity in the source and target domain. We propose to use an entire unlabeled corpus as pivot: this corpus must be general enough to encapsulate the source and target domains of interest. The idea is to (i) learn semantic similarity between words on the pivot corpus and (ii) use tree kernels embedding such a similarity to learn a RE system on the source, which allows to generalize to the new target domain. This reasoning is related to Structural Correspondence Learning (SCL) (Blitzer et al., 2006). In SCL, a representation shared across domains is learned by exploiting pivot features, where a set of pivot features has to be selected (usually a few thousands). In our case *pivots* are words that cooccur with the target words in a large unlabeled corpus and are thus implicitly represented in the similarity matrix. Thus, in contrast to SCL, we do not need to select a set of pivot features but rather rely on the distributional hypothesis to infer a semantic similarity from a large unlabeled corpus. Then, this similarity is incorporated into the tree kernel that provides the necessary restriction for an effective semantic similarity calculation. One peculiarity of our work is that we exploit a large amount of general data, i.e. data gathered from the web, which is a different but also more challenging scenario than the general unsupervised DA setting where domain specific data is available. We study two ways for term generalization in tree kernels: Brown words clusters and Latent Semantic Analysis (LSA), both briefly described next.

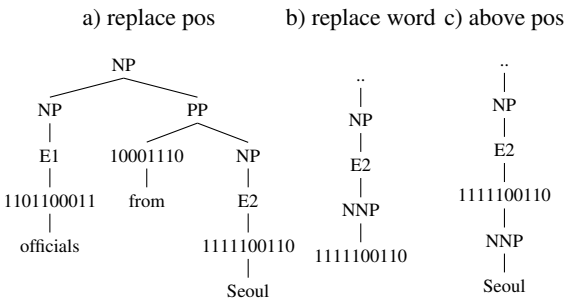

Figure 2: Integrating Brown cluster information

The Brown algorithm (Brown et al., 1992) is a hierarchical agglomerative hard-clustering algorithm. The path from the root of the tree down to a leaf node is represented compactly as a bitstring. By cutting the hierarchy at different levels one can obtain different granularities of word clusters. We

[3] In a setup where gold label info is included, the impact of similarity-based methods is limited – gold information seems to predominate. We argue that whenever gold data is not available, distributional semantics paired with kernels can be useful to improve generalization and complement missing gold info.

evaluate different ways to integrate cluster information into tree kernels, some of which are illustrated in Figure 2.

For LSA, we compute term similarity functions following the *distributional hypothesis* (Harris, 1964), i.e. the meaning of a word can be described by the set of textual contexts in which it appears. The original word-by-word context matrix M is decomposed through Singular Value Decomposition (SVD) (Golub and Kahan, 1965), where M is approximated by $U_l S_l V_l^T$. This approximation supplies a way to project a generic term w_i into the l-dimensional space using $W = U_l S_l^{1/2}$, where each row corresponds to the vectors $\vec{w}_i$. Given two words w_1 and w_2, the term similarity function σ is estimated as the cosine similarity between the corresponding projections $\vec{w}_1, \vec{w}_2$ and used in the kernel as described in Section 2.

5 Experimental Setup

We treat relation extraction as a multi-class classification problem and use SVM-light-TK[4] to train the binary classifiers. The output of the classifiers is combined using the one-vs-all approach. We modified the SVM-light-TK package to include the semantic tree kernels and instance weighting. The entire software package is publicly available.[5] For the SVMs, we use the same parameters as Zhang et al. (2006): $\lambda = 0.4, c = 2.4$ using the Collins Kernel (Collins and Duffy, 2001). The precision/recall trade-off parameter for the *none* class was found on held-out data: $j = 0.2$. Evaluation metrics are standard micro average Precision, Recall and balanced Fscore (F1). To compute statistical significance, we use the approximate randomization test (Noreen, 1989).[6] In all our experiments, we model argument order of the relations explicitly. Thus, for instance for the 7 coarse ACE 2004 relations, we build 14 coarse-grained classifiers (two for each coarse ACE 2004 relation type except for PER-SOC, which is symmetric, and one classifier for the *none* relation).

Data We use two datasets. To compare our model against the state of the art we use the *ACE 2004* data. It contains 348 documents and 4,374 positive relation instances. To generate the training data, we follow prior studies and extract an instance for every pair of mentions in the same sentence, which are separated by no more than three other mentions (Zhang et al., 2006; Sun et al., 2011). After data preprocessing, we obtained 4,327 positive and 39,120 negative instances.

ACE 2005	docs	sents	ASL	relations
nw+bn	298	5029	18.8	3562
bc	52	2267	16.3	1297
cts	34	2696	15.3	603
wl	114	1697	22.6	677

Table 1: Overview of the ACE 2005 data.

For the domain adaptation experiments we use the *ACE 2005* corpus. An overview of the data is given in Table 1. Note that this data is different from ACE 2004: it covers different years (ACE 2004: texts from 2001-2002; ACE 2005: 2003-2005). Moreover, the annotation guidelines have changed (for example, ACE 2005 contains no discourse relation, some relation (sub)types have changed/moved, and care must be taken for differences in SGM markup, etc.).

More importantly, the ACE 2005 corpus covers additional domains: weblogs, telephone conversation, usenet and broadcast conversation. In the experiments, we use news (the union of nw and bn) as source domain, and weblogs (wl), telephone conversations (cts) and broadcast conversation (bc) as target domains.[7] We take half of bc as only target development set, and leave the remaining data and domains for final testing (since they are already small, cf. Table 1). To get a feeling of how these domains differ, Figure 3 depicts the distribution of relations in each domain and Table 2 provides the most frequent out-of-vocabulary words together with their percentage.

Lexical Similarity and Clustering We applied LSA to ukWaC (Baroni et al., 2009), a 2 billion word corpus constructed from the Web[8] using the `s-space` toolkit.[9] Dimensionality reduction was performed using SVD with 250 dimensions, following (Croce et al., 2011). The co-occurrence matrix was transformed by *tfidf*. For the Brown word clusters, we used Percy Liang's implementation[10] of the Brown clustering algorithm (Liang, 2005). We incorporate cluster information by us-

[4] http://disi.unitn.it/moschitti/Tree-Kernel.htm
[5] http://disi.unitn.it/ikernels/RelationExtraction
[6] http://www.nlpado.de/~sebastian/software/sigf.shtml
[7] We did not consider the usenet subpart, since it is among the smaller domains and data-preprocessing was difficult.
[8] http://wacky.sslmit.unibo.it/
[9] http://code.google.com/p/airhead-research/
[10] https://github.com/percyliang/brown-cluster

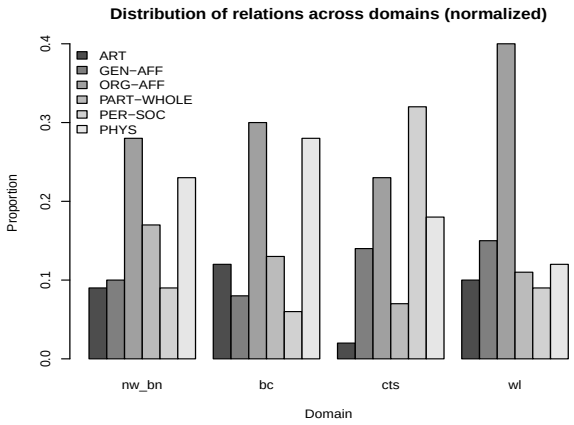

Figure 3: Distribution of relations in ACE 2005.

Dom	Most frequent OOV words
bc (24%)	insurance, unintelligible, malpractice, ph, clip, colonel, crosstalk
cts (34%)	uh, Yeah, um, eh, mhm, uh-huh, ˜, ah, mm, th, plo, topic, y, workplace
wl (49%)	title, Starbucks, Well, blog, !!, werkheiser, undefeated, poor, shit

Table 2: For each domain the percentage of target domain words (types) that are unseen in the source together with the most frequent OOV words.

ing the 10-bit cluster prefix (Sun et al., 2011; Chan and Roth, 2010). For the domain adaptation experiments, we use ukWaC corpus-induced clusters as bridge between domains. We limited the vocabulary to that in ACE 2005, which are approximately 16k words. Following previous work, we left case intact in the corpus and induced 1,000 word clusters from words appearing at least 100 times.[11]

DA baseline We compare our approach to instance weighting (Jiang and Zhai, 2007). We modified SVM-light-TK such that it takes a parameter vector $\beta_i, .., \beta_m$ as input, where each β_i represents the relative *importance* of example i with respect to the target domain (Huang et al., 2007; Widmer, 2008). To estimate the importance weights, we train a binary classifier that distinguishes between source and target domain instances. We consider the union of the three target domains as target data. To train the classifier, the source instances are marked as negative and the target instances are marked as positive. Then, this classifier is applied to the source data. To obtain the weights β_i, we convert the SVM scores into posterior probabilities by training a sigmoid using the modified Platt algorithm (Lin et al., 2007).[12]

[11] Clusters are available at http://disi.unitn.it/ikernels/RelationExtraction

Prior Work:	Type	P	R	F1
Zhang (2006), tree only	K,yes	74.1	62.4	67.7
Zhang (2006), linear	K,yes	73.5	67.0	70.1
Zhang (2006), poly	K,yes	76.1	68.4	72.1
Sun & Grishman (2011)	F,yes	73.4	67.7	70.4
Jiang & Zhai (2007)	F,no	73.4	70.2	71.3
Our re-implementation:	**Type**	**P**	**R**	**F1**
Tree only (PET Zhang)	K,yes	70.7	62.5	66.3
Linear composite	K,yes	71.3	66.6	68.9
Polynomial composite	K,yes	72.6	67.7	70.1

Table 3: Comparison to previous work on the 7 relations of ACE 2004. K: kernel-based; F: feature-based; yes/no: models argument order explicitly.

6 Results

6.1 Alignment to Prior Work

Although most prior studies performed 5-fold cross-validation on ACE 2004, it is often not clear whether the partitioning has been done on the instance or on the document level. Moreover, it is often not stated whether argument order is modeled explicitly, making it difficult to compare system performance. Citing Wang (2008), "We feel that there is a sense of increasing confusion down this line of research". To ease comparison for future research we use the same 5-fold split on the document level as Sun et al. (2011)[13] and make our system publicly available (see Section 5).

Table 3 shows that our system (bottom) aligns well with the state of the art. Our best system (composite kernel with polynomial expansion) reaches an F1 of 70.1, which aligns well to the 70.4 of Sun et al. (2011) that use the same data-split. This is slightly behind that of Zhang (2006); the reason might be threefold: i) different data partitioning; ii) different pre-processing; iii) they incorporate features from additional sources, i.e. a phrase chunker, dependency parser and semantic resources (Zhou et al., 2005) (we have on average 9 features/instance, they use 40). Since we focus on evaluating the impact of semantic similarity in tree kernels, we think our system is very competitive. Removing gold entity and mention

[12] Other weightings/normalizations (like LDA) didn't improve the results; best was to take the posteriors and add c.

[13] http://cs.nyu.edu/~asun/pub/ACL11_CVFileList.txt

information results in a significant F1 drop from 66.3% to 54.2%. However, in a realistic setting we do not have gold entity info available, especially not in the case when we apply the system *to any kind of text*. Thus, in the domain adaptation setup we assume entity boundaries given but not their label. Clearly, evaluating the approach on predicted mentions, e.g. Giuliano et al. (2007), is another important dimension, however, out of the scope of the current paper.

6.2 Tree Kernels with Brown Word Clusters

To evaluate the effectiveness of Brown word clusters in tree kernels, we evaluated different instance representations (cf. Figure 2) on the ACE 2005 development set. Table 4 shows the results.

bc-dev	P	R	F1
baseline	52.2	41.7	46.4
replace word	49.7	38.6	43.4
replace pos	56.3	41.9	**48.0**
replace pos only mentions	55.3	41.6	47.5
above word	54.5	42.2	47.6
above pos	55.8	41.1	47.3

Table 4: Brown clusters in tree kernels (cf. Fig 2).

To summarize, we found: i) it is generally a bad idea to dismiss lexical information completely, i.e. replacing or ignoring terminals harms performance; ii) the best way to incorporate Brown clusters is to replace the Pos tag with the cluster bitstring; iii) marking all words is generally better than only mentions; this is in contrast to Sun et al. (2011) who found that in their feature-based system it was better to add cluster information to entity mentions only. As we will discuss, the combination of syntax and semantics exploited in this novel kernel avoids the necessity of restricting cluster information to mentions only.

6.3 Semantic Tree Kernels for DA

To evaluate the effectiveness of the proposed kernels across domains, we use the ACE 2005 data as testbed. Following standard practices on ACE 2004, the newswire (nw) and broadcast news (bn) data from ACE 2005 are considered training data (labeled source domain). The test data consists of three targets: broadcast conversation, telephone conversation, weblogs. As we want to build a single system that is able to handle heterogeneous data, we do not assume that there is further unlabeled domain-specific data, but we assume to have a large unlabeled corpus (ukWaC) at our disposal to improve the generalizability of our models.

Table 5 presents the results. In the first three rows we see the performance of the baseline models (PET, BOW and BOW without marking). In-domain (col 1): when evaluated on the same domain the system was trained on (nw+bn, 5-fold cross-validation). Out-of-domain performance (cols 2-4): the system evaluated on the targets, namely broadcast conversation (bc), telephone conversation (cts) and weblogs (wl). While the system achieves a performance of 46.0 F1 within its own domain, the performance drops to 45.3, 43.4 and 34.0 F1 on the target domains, respectively. The BOW kernel that disregards syntax is often less effective (row 2). We see also the effect of target entity marking: the BOW kernel without entity marking performs substantially worse (row 3). For the remaining experiments we use the BOW kernel with entity marking.

Rows 4 and 5 of Table 5 show the effect of using instance weighting for the PET baseline. Two models are shown: they differ in whether PET or BOW was used as instance representation for training the discriminative classifier. Instance weighting shows mixed results: it helps slightly on the weblogs domain, but does not help on broadcast conversation and telephone conversations. Interestingly, the two models used to obtain the weights perform similarly, despite the fact that their performance differs (F1: 70.5 BOW, 73.5 PET); it turns out that the correlation between the weights is high (+0.82).

The next part (rows 6-9) shows the effect of enriching the syntactic structures with either Brown word clusters or LSA. The Brown cluster kernel applied to PET (P_WC) improves performance over the baseline over *all* target domains. The same holds also for the lexical semantic kernel based on LSA (P_LSA), however, to only two out of three domains. This suggests that the two kernels capture different information and a combined kernel might be effective. More importantly, the table shows the effect of adding Brown clusters or LSA semantics to the BOW kernel: it can actually hurt performance, sometimes to a small but other times to a considerably degree. For instance, WC applied to PET achieves an F1 of 47.0 (baseline: 45.3) on the bc domain, while applied to BOW it hurts performance significantly, i.e. it drops from

	nw+bn (in-dom.)			**bc**			**cts**			**wl**		
Baseline:	P:	R:	F1:	P:	R:	F1:	P:	R:	F1:	P:	R:	F1:
PET	50.6	42.1	46.0	51.2	40.6	45.3	51.0	37.8	43.4	35.4	32.8	34.0
BOW	55.1	37.3	44.5	57.2	37.1	45.0	57.5	31.8	41.0	41.1	27.2	32.7
BOW no marking	49.6	34.6	40.7	51.5	34.7	41.4	54.6	30.7	39.3	37.6	25.7	30.6
PET adapted:	P:	R:	F:	P:	R:	F:	P:	R:	F:	P:	R:	F:
IW1 (using PET)	51.4	44.1	47.4	49.1	41.1	44.7	50.8	37.5	43.1	35.5	33.9	34.7
IW2 (using BOW)	51.2	43.6	47.1	49.1	41.3	44.9	51.2	37.8	43.5	35.6	33.8	34.7
With Similarity:	P:	R:	F1:	P:	R:	F1:	P:	R:	F1:	P:	R:	F1:
P_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
B_WC	47.9	36.4	41.4	49.5	35.2	41.2	53.3	33.2	40.9	31.7	24.1	27.4
P_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
B_LSA	53.7	37.8	44.4	55.1	33.8	41.9	54.9	32.3	40.7	39.2	28.6	33.0
P+P_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
P+P_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
P+P_WC+P_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†

Table 5: In-domain (first column) and out-of-domain performance (columns two to four) on ACE 2005. PET and BOW are abbreviated by P and B, respectively. If not specified BOW is *marked*.

45.0 to 41.2. This is also the case for LSA applied to the BOW kernel, which drops to 41.9. On the cts domain this is less pronounced. Only on the weblogs domain B_LSA achieves a minor improvement (from 32.7 to 33.0). In general, distributional semantics constrained by syntax (i.e. combined with PET) can be effectively exploited, while if applied 'blindly' – without the guide of syntax (i.e. BOW) – performance might drop, often considerably. We believe that the semantic information does not help the BOW kernel as there is no syntactic information that constrains the application of the noisy source, as opposed to the case with the PET kernel.

As the two semantically enriched kernels, PET_LSA and PET_WC, seem to capture different information we use composite kernels (rows 10-11): the baseline kernel (PET) summed with the lexical semantic kernels. As we can see, results improve further: for instance on the bc test set, PET_WC reaches an F1 of 47.0, while combined with PET (PET+PET_WC) this improves to 48.3. Adding also PET_LSA results in the best performance and our final system (last row): the composite kernel (PET+PET_WC+PET_LSA) reaches an F1 of 48.5, 43.6 and 37.8 on the target domains, respectively, i.e. with an absolute improvement of: +3.2%, +0.2% and +3.8%, respectively. Two out of three improvements are significant at $p < 0.05$ (indicated by † in Table 5). Moreover, the system also improved in its own domain (first column), therefore having achieved robustness.

By performing an error analysis we found that, for instance, the Brown clusters help to generalize locations and professions. For example, the baseline incorrectly considered 'Dutch filmmaker' in a PART-WHOLE relation, while our system correctly predicted GEN-AFF(filmmaker,Dutch). 'Filmmaker' does not appear in the source, however 'Dutch citizen' does. Both 'citizen' and 'filmmaker' appear in the same cluster, thereby helping the system to recover the correct relation.

	bc		**cts**		**wl**	
Relation:	BL	SYS	BL	SYS	BL	SYS
PART-WHOLE	37.8	**43.1**	**59.3**	52.3	30.5	**36.3**
ORG-AFF	60.7	**62.9**	35.5	**42.3**	41.0	**42.0**
PHYS	35.3	**37.6**	25.4	**28.7**	25.2	**26.9**
ART	20.8	**37.9**	34.5	**43.5**	26.5	**40.3**
GEN-AFF	30.1	**33.0**	16.8	**18.6**	21.6	**28.1**
PER-SOC	74.1	**74.2**	**66.3**	63.1	42.6	**48.0**
μ average	45.3	**48.5**	43.4	**43.6**	34.0	**37.8**

Table 6: F1 per coarse relation type (ACE 2005). SYS is the final model, i.e. last row (PET+PET_WC+PET_LSA) of Table 5.

Furthermore, Table 6 provides the performance breakdown per relation for the baseline (BL) and our best system (SYS). The table shows that our system is able to improve F1 on *all* relations for the broadcast and weblogs data. On most relations, this is also the case for the telephone (cts) data, although the overall improvement is not significant. Most errors were made on the PER-SOC

relation, which constitutes the largest portion of cts (cf. Figure 3). As shown in the same figure, the relation distribution of the cts domain is also rather different from the source. This conversation data is a very hard domain, with a lot of disfluencies and spoken language patterns. We believe it is more distant from the other domains, especially from the unlabeled collection, thus other approaches might be more appropriate, e.g. domain identification (Dredze et al., 2010).

7 Conclusions and Future Work

We proposed syntactic tree kernels enriched by lexical semantic similarity to tackle the portability of a relation extractor to different domains. The results of diverse kernels exploiting (i) Brown clustering and (ii) LSA show that a suitable combination of syntax and lexical generalization is very promising for domain adaptation. The proposed system is able to improve performance significantly on two out of three target domains (up to 8% relative improvement). We compared it to instance weighting, which gave only modest or no improvements. Brown clusters remained unexplored for kernel-based approaches. We saw that adding cluster information blindly might actually hurt performance. In contrast, adding lexical information combined with syntax can help to improve performance: the syntactic structure enriched with lexical information provides a feature space where syntax constrains lexical similarity obtained from unlabeled data. Thus, semantic syntactic tree kernels appear to be a suitable mechanism to adequately trade off the two kinds of information. In future we plan to extend the evaluation to predicted mentions, which necessarily includes a careful evaluation of pre-processing components, as well as evaluating the approach on other semantic tasks.

Acknowledgments

We would like to thank Min Zhang for discussions on his prior work as well as the anonymous reviewers for their valuable feedback. The research described in this paper has been supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under the grant #288024: LIMOSINE – Linguistically Motivated Semantic aggregation engiNes.

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Language Resources and Evaluation*, pages 209–226.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain Adaptation with Structural Correspondence Learning. In *Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia.

Stephan Bloehdorn and Alessandro Moschitti. 2007a. Combined syntactic and semantic kernels for text classification. In *ECIR*, pages 307–318.

Stephan Bloehdorn and Alessandro Moschitti. 2007b. Exploiting Structure and Semantics for Expressive Text Kernels. In *Conference on Information Knowledge and Management*, Lisbon, Portugal.

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. *Computational Linguistics*, 18:467–479.

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

Yee Seng Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 152–160, Beijing, China, August. Coling 2010 Organizing Committee.

Michael Collins and Nigel Duffy. 2001. Convolution Kernels for Natural Language. In *Proceedings of Neural Information Processing Systems (NIPS 2001)*.

Danilo Croce, Alessandro Moschitti, and Roberto Basili. 2011. Semantic convolution kernels over dependency trees: smoothed partial tree kernel. In *CIKM*, pages 2013–2016.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on ACL*, Barcelona, Spain.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of ACL*, pages 256–263, Prague, Czech Republic, June.

Mark Dredze, Tim Oates, and Christine Piatko. 2010. We're not in kansas anymore: Detecting domain changes in streams. In *Proceedings of EMNLP*, pages 585–595, Cambridge, MA.

Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. 2007. Relation extraction and the influence of automatic named-entity recognition. *ACM Trans. Speech Lang. Process.*, 5(1):2:1–2:26, December.

G. Golub and W. Kahan. 1965. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*, 2(2):pp. 205–224.

Zellig Harris. 1964. Distributional structure. In Jerrold J. Katz and Jerry A. Fodor, editors, *The Philosophy of Linguistics*. Oxford University Press.

Jiayuan Huang, Arthur Gretton, Bernhard Schölkopf, Alexander J. Smola, and Karsten M. Borgwardt. 2007. Correcting sample selection bias by unlabeled data. In *In NIPS*. MIT Press.

Jing Jiang and Chengxiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *In ACL 2007*, pages 264–271.

Jing Jiang. 2009. Multi-task transfer learning for weakly-supervised relation extraction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th IJCNLP*, pages 1012–1020, Suntec, Singapore.

Percy Liang. 2005. Semi-Supervised Learning for Natural Language. Master's thesis, Massachusetts Institute of Technology.

Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng. 2007. A note on platt's probabilistic outputs for support vector machines. *Mach. Learn.*, 68(3):267–276.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, pages 419–444.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-IJCNLP*, pages 1003–1011, Suntec, Singapore, August.

Alessandro Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *Proceedings of the 42nd Meeting of the ACL*, Barcelona, Spain.

Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of the 17th ECML*, Berlin, Germany.

Alessandro Moschitti. 2008. Kernel methods, syntax and semantics for relational text categorization. In *CIKM*, pages 253–262.

Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of EMNLP '09*, pages 1378–1387, Stroudsburg, PA, USA.

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL*, pages 915–932.

Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley-Interscience.

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

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Anders Søgaard and Martin Haulrich. 2011. Sentence-level instance-weighting for graph-based and transition-based dependency parsing. In *Proceedings of the 12th International Conference on Parsing Technologies*, IWPT '11, pages 43–47, Stroudsburg, PA, USA.

Anders Søgaard and Anders Johannsen. 2012. Robust learning in random subspaces: equipping NLP for OOV effects. In *Proceedings of Coling*.

Ang Sun, Ralph Grishman, and Satoshi Sekine. 2011. Semi-supervised relation extraction with large-scale word clustering. In *Proceedings of ACL-HLT*, pages 521–529, Portland, Oregon, USA.

Mengqiu Wang. 2008. A re-examination of dependency path kernels for relation extraction. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing-IJCNLP*.

Christian Widmer. 2008. Domain adaptation in sequence analysis. Diplomarbeit, University of Tübingen.

Feiyu Xu, Hans Uszkoreit, Hond Li, and Niko Felger. 2008. Adaptation of relation extraction rules to new domains. In *Proceedings of LREC'08*, Marrakech, Morocco.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2002. Kernel methods for relation extraction. In *Proceedings of EMNLP-ACL*, pages 181–201.

Min Zhang, Jian Su, Danmei Wang, Guodong Zhou, and Chew Lim Tan. 2005. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *Proceedings of IJCNLP'2005*, pages 378–389, Jeju Island, South Korea.

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 *Proceedings of COLING-ACL 2006*, pages 825–832.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of ACL)*, pages 427–434, Ann Arbor, Michigan.