Decomposition Kernels for Natural Language Processing

Fabrizio Costa **Sauro Menchetti** **Alessio Ceroni** **Andrea Passerini** **Paolo Frasconi**
Dipartimento di Sistemi e Informatica,
Università degli Studi di Firenze,
via di S. Marta 3, 50139 Firenze, Italy
{costa,menchett,passerini,aceroni,p-f} AT dsi.unifi.it

Abstract

We propose a simple solution to the sequence labeling problem based on an extension of weighted decomposition kernels. We additionally introduce a multi-instance kernel approach for representing lexical word sense information. These new ideas have been preliminarily tested on named entity recognition and PP attachment disambiguation. We finally suggest how these techniques could be potentially merged using a declarative formalism that may provide a basis for the integration of multiple sources of information when using kernel-based learning in NLP.

1 Introduction

Many tasks related to the analysis of natural language are best solved today by machine learning and other data driven approaches. In particular, several subproblems related to information extraction can be formulated in the supervised learning framework, where statistical learning has rapidly become one of the preferred methods of choice. A common characteristic of many NLP problems is the relational and structured nature of the representations that describe data and that are internally used by various algorithms. Hence, in order to develop effective learning algorithms, it is necessary to cope with the inherent structure that characterize linguistic entities. Kernel methods (see e.g. Shawe-Taylor and Cristianini, 2004) are well suited to handle learning tasks in structured domains as the statistical side of a learning algorithm can be naturally decoupled from any representational details that are handled by the kernel function. As a matter of facts, kernel-based statistical learning has gained substantial importance in the NLP field. Applications are numerous and diverse and include for example refinement of statistical parsers (Collins and Duffy, 2002), tagging named entities (Cumby and Roth, 2003; Tsochantaridis et al., 2004), syntactic chunking (Daumé III and Marcu, 2005), extraction of relations between entities (Zelenko et al., 2003; Culotta and Sorensen, 2004), semantic role labeling (Moschitti, 2004). The literature is rich with examples of kernels on discrete data structures such as sequences (Lodhi et al., 2002; Leslie et al., 2002; Cortes et al., 2004), trees (Collins and Duffy, 2002; Kashima and Koyanagi, 2002), and annotated graphs (Gärtner, 2003; Smola and Kondor, 2003; Kashima et al., 2003; Horváth et al., 2004). Kernels of this kind can be almost invariably described as special cases of convolution and other decomposition kernels (Haussler, 1999). Thanks to its generality, decomposition is an attractive and flexible approach for defining the similarity between structured objects starting from the similarity between smaller parts. However, excessively large feature spaces may result from the combinatorial growth of the number of distinct subparts with their size. When too many dimensions in the feature space are irrelevant, the Gram matrix will be nearly diagonal (Schölkopf et al., 2002), adversely affecting generalization in spite of using large margin classifiers (Ben-David et al., 2002). Possible cures include extensive use of prior knowledge to guide the choice of relevant parts (Cumby and Roth, 2003; Frasconi et al., 2004), the use of feature selection (Suzuki et al., 2004), and soft matches (Saunders et al., 2002). In (Menchetti et al., 2005) we have shown that better generalization can indeed be achieved by avoiding hard comparisons between large parts. In a weighted decomposition kernel (WDK) only small parts are matched, whereas the importance of the match is determined by comparing the sufficient statistics of elementary probabilistic models fitted on larger contextual substructures. Here we introduce a position-dependent version of WDK that can solve sequence labeling problems without searching the output space, as required by other recently proposed kernel-based solutions (Tsochantaridis et al., 2004; Daumé III and Marcu, 2005).

The paper is organized as follows. In the next two sections we briefly review decomposition kernels and its weighted variant. In Section 4 we introduce a version of WDK for solving supervised sequence labeling tasks and report a preliminary evaluation on a named entity recognition problem. In Section 5 we suggest a novel multi-instance approach for representing WordNet information and present an application to the PP attachment ambiguity resolution problem. In Section 6 we discuss how these ideas could be merged using a declarative formalism in order to integrate multiple sources of information when using kernel-based learning in NLP.

2 Decomposition Kernels

An R-decomposition structure (Haussler, 1999; Shawe-Taylor and Cristianini, 2004) on a set $\mathcal{X}$ is a triple $\mathcal{R} = \langle \vec{\mathcal{X}}, R, \vec{k} \rangle$ where $\vec{\mathcal{X}} = (\mathcal{X}_1, \ldots, \mathcal{X}_D)$ is a D–tuple of non–empty subsets of $\mathcal{X}$, R is a finite relation on $\mathcal{X}_1 \times \cdots \times \mathcal{X}_D \times \mathcal{X}$, and $\vec{k} = (k_1, \ldots, k_D)$ is a D–tuple of positive definite kernel functions $k_d : \mathcal{X}_d \times \mathcal{X}_d \mapsto \mathbb{R}$. $R(\vec{x}, x)$ is true iff $\vec{x}$ is a tuple of "parts" for x — i.e. $\vec{x}$ is a decomposition of x. Note that this definition of "parts" is very general and does not require the parthood relation to obey any specific mereological axioms, such as those that will be introduced in Section 6. For any $x \in \mathcal{X}$, let $R^{-1}(x) = \{(x_1, \ldots, x_D) \in \vec{\mathcal{X}} : R(\vec{x}, x)\}$ denote the multiset of all possible decompositions[1] of x. A decomposition kernel is then defined as the *multiset kernel* between the decompositions:

$$K_{\mathcal{R}}(x, x') = \sum_{\substack{\vec{x} \in R^{-1}(x) \\ \vec{x}' \in R^{-1}(x')}} \prod_{d=1}^{D} \kappa_d(x_d, x'_d) \tag{1}$$

where, as an alternative way of combining the kernels, we can use the product instead of a summation: intuitively this increases the feature space dimension and makes the similarity measure more selective. Since decomposition kernels form a rather vast class, the relation R needs to be carefully tuned to different applications in order to characterize a suitable kernel. As discussed in the Introduction, however, taking *all* possible subparts into account may lead to poor predictivity because of the combinatorial explosion of the feature space.

3 Weighted Decomposition Kernels

A weighted decomposition kernel (WDK) is characterized by the following decomposition structure:

$$\mathcal{R} = \langle \vec{\mathcal{X}}, R, (\delta, \kappa_1, \ldots, \kappa_D) \rangle$$

where $\vec{\mathcal{X}} = (S, Z_1, \ldots, Z_D)$, $R(s, z_1, \ldots, z_D, x)$ is true iff $s \in S$ is a subpart of x called the *selector* and $\vec{z} = (z_1, \ldots, z_D) \in Z_1 \times \cdots \times Z_D$ is a tuple of subparts of x called the *contexts* of s in x. Precise definitions of s and $\vec{z}$ are domain-dependent. For example in (Menchetti et al., 2005) we present two formulations, one for comparing whole sequences (where both the selector and the context are subsequences), and one for comparing attributed graphs (where the selector is a single vertex and the context is the subgraph reachable from the selector within a short path). The definition is completed by introducing a kernel on selectors and a kernel on contexts. The former can be chosen to be the exact matching kernel, δ, on $S \times S$, defined as $\delta(s, s') = 1$ if $s = s'$ and $\delta(s, s') = 0$ otherwise. The latter, κ_d, is a kernel on $Z_d \times Z_d$ and provides a soft similarity measure based on attribute frequencies. Several options are available for context kernels, including the discrete version of probability product kernels (PPK) (Jebara et al., 2004) and histogram intersection kernels (HIK) (Odone et al., 2005). Assuming there are n categorical attributes, each taking on m_i distinct values, the context kernel can be defined as:

$$\kappa_d(z, z') \quad = \quad \sum_{i=1}^{n} k_i(z, z') \tag{2}$$

where k_i is a kernel on the i-th attribute. Denote by $p_i(j)$ the observed frequency of value j in z. Then

[1] Decomposition examples in the string domain include taking all the contiguous fixed-length substrings or all the possible ways of dividing a string into two contiguous substrings.

k_i can be defined as a HIK or a PPK respectively:

$$k_i(z, z') = \sum_{j=1}^{m_i} \min\{p_i(j), p'_i(j)\} \quad (3)$$

$$k_i(z, z') = \sum_{j=1}^{m_i} \sqrt{p_i(j) \cdot p'_i(j)} \quad (4)$$

This setting results in the following general form of the kernel:

$$K(x, x') = \sum_{\substack{(s, \vec{z}) \in R^{-1}(x) \\ (s', \vec{z}') \in R^{-1}(x')}} \delta(s, s') \sum_{d=1}^{D} \kappa_d(z_d, z'_d) \quad (5)$$

where we can replace the summation of kernels with $\prod_{d=1}^{D} 1 + \kappa_d(z_d, z'_d)$.

Compared to kernels that simply count the number of substructures, the above function weights different matches between selectors according to contextual information. The kernel can be afterwards normalized in $[-1, 1]$ to prevent similarity to be boosted by the mere size of the structures being compared.

4 WDK for sequence labeling and applications to NER

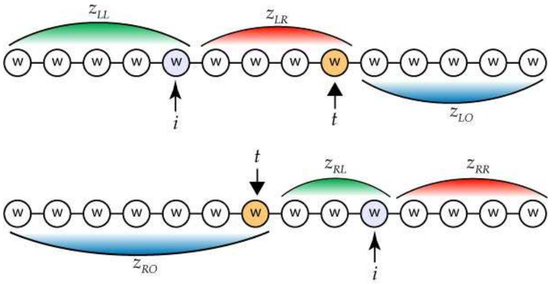
In a sequence labeling task we want to map input sequences to output sequences, or, more precisely, we want to map each element of an input sequence that takes label from a source alphabet to an element with label in a destination alphabet.

Here we cast the sequence labeling task into position specific classification, where different sequence positions give independent examples. This is different from previous approaches in the literature where the sequence labeling problem is solved by searching in the output space (Tsochantaridis et al., 2004; Daumé III and Marcu, 2005). Although the method lacks the potential for collectively labeling all positions simultaneously, it results in a much more efficient algorithm.

In the remainder of the section we introduce a specialized version of the weighted decomposition kernel suitable for a sequence transduction task originating in the natural language processing domain: the named entity recognition (NER) problem, where we map sentences to sequences of a reduced number of named entities (see Sec.4.1).

More formally, given a finite dictionary Σ of words and an input sentence $x \in \Sigma^*$, our input objects are pairs of sentences and indices $r = (x, t)$ where $r \in \Sigma^* \times \mathbb{N}$. Given a sentence x, two integers $b \geq 1$ and $b \leq e \leq |x|$, let $x[b]$ denote the word at position b and $x[b..e]$ the sub-sequence of x spanning positions from b to e. Finally we will denote by $\xi(x[b])$ a word attribute such as a morphological trait (*is a number* or *has capital initial*, see 4.1) for the word in sentence x at position b.

Figure 1: Sentence decomposition.



We introduce two versions of WDK: one with four context types ($D = 4$) and one with increased contextual information ($D = 6$) (see Eq. 5). The relation R depends on two integers t and $i \in \{1, \ldots, |x|\}$, where t indicates the position of the word we want to classify and i the position of a generic word in the sentence. The relation for the first kernel version is defined as: $R = \{(s, z_{LL}, z_{LR}, z_{RL}, z_{RR}, r)\}$ such that the selector $s = x[i]$ is the word at position i, the z_{LL} (LeftLeft) part is a sequence defined as $x[1..i]$ if $i < t$ or the null sequence ε otherwise and the z_{LR} (LeftRight) part is the sequence $x[i + 1..t]$ if $i < t$ or ε otherwise. Informally, z_{LL} is the initial portion of the sentence up to word of position i, and z_{LR} is the portion of the sentence from word at position $i + 1$ up to t (see Fig. 1). Note that when we are dealing with a word that lies to the left of the target word t, its z_{RL} and z_{RR} parts are empty. Symmetrical definitions hold for z_{RL} and z_{RR} when $i > t$. We define the weighted decomposition kernel for sequences as

$$K(r, r') = \sum_{t=1}^{|x|} \sum_{t'=1}^{|x'|} \delta_\xi(s, s') \sum_{d \in \{LL, LR, RL, RR\}} \kappa(z_d, z'_d) \quad (6)$$

where $\delta_\xi(s, s') = 1$ if $\xi(s) = \xi(s')$ and 0 otherwise (that is δ_ξ checks whether the two selector words have the same morphological trait) and κ is Eq. 2 with only one attribute which then boils down to Eq. 3 or Eq. 4, that is a kernel over the histogram for word occurrences over a specific part.

Intuitively, when applied to word sequences, this kernel considers separately words to the left of the entry we want to transduce and those to its right. The kernel computes the similarity for each sub-sequence by matching the corresponding bag of enriched words: each word is matched only with words that have the same trait as extracted by ξ and the match is then weighted proportionally to the frequency count of identical words preceding and following it.

The kernel version with D=6 adds two parts called z_{LO} (LeftOther) and z_{RO} (RightOther) defined as $x[t+1..|r|]$ and $x[1..t]$ respectively; these represent the remaining sequence parts so that $x = z_{LL} \circ z_{LR} \circ z_{LO}$ and $x = z_{RL} \circ z_{RR} \circ z_{RO}$.

Note that the WDK transforms the sentence in a bag of enriched words computed in a preprocessing phase thus achieving a significant reduction in computational complexity (compared to the recursive procedure in (Lodhi et al., 2002)).

4.1 Named Entity Recognition Experimental Results

Named entities are phrases that contain the names of persons, organizations, locations, times and quantities. For example in the following sentence:

```
[PER Wolff ] , currently a journalist in [LOC
Argentina ] , played with [PER Del Bosque ] in the
final years of the seventies in [ORG Real Madrid].
```

we are interested in predicting that Wolff and Del Bosque are people's names, that Argentina is a name of a location and that Real Madrid is a name of an organization.

The chosen dataset is provided by the shared task of CoNLL–2002 (Saunders et al., 2002) which concerns language–independent named entity recognition. There are four types of phrases: person names (PER), organizations (ORG), locations (LOC) and miscellaneous names (MISC), combined with two tags, B to denote the first item of a phrase and I for any non–initial word; all other phrases are classified as (OTHER). Of the two available languages (Spanish and Dutch), we run experiments only on the Spanish dataset which is a collection of news wire articles made available by the Spanish EFE News Agency. We select a subset of 300 sentences for training and we evaluate the performance on test set. For each category, we evaluate the $F_{\beta=1}$ measure of 4 versions of WDK: word histograms are matched using HIK (Eq. 3) and the kernels on various parts (z_{LL}, z_{LR},etc) are combined with a summation SumHIK or product ProHIK; alternatively the histograms are combined with a PPK (Eq. 4) obtaining SumPPK, ProPPK.

Table 1: NER experiment D=4

Class	SumHis	ProHis	SumPro	ProPro
B-LOC	74.33	68.68	72.12	66.47
I-LOC	58.18	52.76	59.24	52.62
B-MISC	52.77	43.31	46.86	39.00
I-MISC	79.98	80.15	77.85	79.65
B-ORG	69.00	66.87	68.42	67.52
I-ORG	76.25	75.30	75.12	74.76
B-PER	60.11	56.60	59.33	54.80
I-PER	65.71	63.39	65.67	60.98
Micro $F_{\beta=1}$	69.28	66.33	68.03	65.30

Table 2: NER experiment with D=6

Class	SumHis	ProHis	SumPro	ProPro
B-LOC	74.81	73.30	73.65	73.69
I-LOC	57.28	58.87	57.76	59.44
B-MISC	56.54	64.11	57.72	62.11
I-MISC	78.74	84.23	79.27	83.04
B-ORG	70.80	73.02	70.48	73.10
I-ORG	76.17	78.70	74.26	77.51
B-PER	66.25	66.84	66.04	67.46
I-PER	68.06	71.81	69.55	69.55
Micro $F_{\beta=1}$	70.69	72.90	70.32	72.38

The word attribute considered for the selector is a word morphologic trait that classifies a word in one of five possible categories: normal word, number, all capital letters, only capital initial and contains non alphabetic characters, while the context histograms are computed counting the exact word frequencies.

Results reported in Tab. 1 and Tab. 2 show that performance is mildly affected by the different choices on how to combine information on the various contexts, though it seems clear that increasing contextual information has a positive influence.

Note that interesting preliminary results can be obtained even without the use of any refined language knowledge, such as part of speech tagging or shallow/deep parsing.

5 Kernels for word semantic ambiguity

Parsing a natural language sentence often involves the choice between different syntax structures that are equally admissible in the given grammar. One of the most studied ambiguity arise when deciding between attaching a prepositional phrase either to the noun phrase or to the verb phrase. An example could be:

1. *eat salad with forks* (attach to verb)
2. *eat salad with tomatoes* (attach to noun)

The resolution of such ambiguities is usually performed by the human reader using its past experiences and the knowledge of the words meaning. Machine learning can simulate human experience by using corpora of disambiguated phrases to compute a decision on new cases. However, given the number of different words that are currently used in texts, there would never be a sufficient dataset from which to learn. Adding semantic information on the possible word meanings would permit the learning of rules that apply to entire categories and can be generalized to all the member words.

5.1 Adding Semantic with WordNet

WordNet (Fellbaum, 1998) is an electronic lexical database of English words built and annotated by linguistic researchers. WordNet is an extensive and reliable source of semantic information that can be used to enrich the representation of a word. Each word is represented in the database by a group of *synonym sets* (synset), with each synset corresponding to an individual linguistic concept. All the synsets contained in WordNet are linked by relations of various types. An important relation connects a synset to its *hypernyms*, that are its immediately broader concepts. The hypernym (and its opposite *hyponym*) relation defines a semantic hierarchy of synsets that can be represented as a directed acyclic graph. The different lexical categories (verbs, nouns, adjectives and adverbs) are contained in distinct hierarchies and each one is rooted by many synsets.

Several metrics have been devised to compute a similarity score between two words using WordNet. In the following we resort to a multiset version of the proximity measure used in (Siolas and d'Alche Buc, 2000), though more refined alternatives are also possible (for example using the conceptual density as in (Basili et al., 2005)). Given the acyclic nature of the semantic hierarchies, each synset can be represented by a group of paths that follows the hypernym relations and finish in one of the top level concepts. Two paths can then be compared by counting how many steps from the roots they have in common. This number must then be normalized dividing by the square root of the product between the path lengths. In this way one can accounts for the unbalancing that arise from different parts of the hierarchies being differently detailed. Given two paths π and π', let l and l' be their lengths and n be the size of their common part, the resulting kernel is:

$$k(\pi, \pi') = \frac{n}{\sqrt{l \cdot l'}} \quad (7)$$

The demonstration that k is positive definite arise from the fact that n can be computed as a positive kernel k^* by summing the exact match kernels between the corresponding positions in π and π' seen as sequences of synset identifiers. The lengths l and l' can then be evaluated as $k^*(\pi, \pi)$ and $k^*(\pi', \pi')$ and k is the resulting normalized version of k^*.

The kernel between two synsets σ and σ' can then be computed by the multi-set kernel (Gärtner et al., 2002a) between their corresponding paths. Synsets are organized into forty-five lexicographer files based on syntactic category and logical groupings. Additional information can be derived by comparing the identifiers λ and λ' of these file associated to σ and σ'. The resulting synset kernel is:

$$\kappa_\sigma(\sigma, \sigma') = \delta(\lambda, \lambda') + \sum_{\pi \in \Pi} \sum_{\pi' \in \Pi'} k(\pi, \pi') \quad (8)$$

where Π is the set of paths originating from σ and the exact match kernel $\delta(\lambda, \lambda')$ is 1 if $\lambda \equiv \lambda'$ and 0 otherwise. Finally, the kernel κ_ω between two words is itself a multi-set kernel between the corresponding sets of synsets:

$$\kappa_\omega(\omega, \omega') = \sum_{\sigma \in \Sigma} \sum_{\sigma' \in \Sigma'} \kappa_\sigma(\sigma, \sigma') \quad (9)$$

where Σ are the synsets associated to the word ω.

5.2 PP Attachment Experimental Results

The experiments have been performed using the Wall-Street Journal dataset described in (Ratnaparkhi et al., 1994). This dataset contains $20,800$ training examples and $3,097$ testing examples. Each phrase x in the dataset is reduced to a verb x_v, its object noun x_{n_1} and prepositional phrase formed by a preposition x_p and a noun x_{n_2}. The target is either V or N whether the phrase is attached to the verb or the noun. Data have been preprocessed by assigning to all the words their corresponding synsets. Additional meanings derived from specific synsets have been attached to the words as described in (Stetina and Nagao, 1997). The kernel between two phrases x and x' is then computed by combining the kernels between single words using either the sum or the product.

Method	Acc	Pre	Rec
S	84.6% ± 0.65%	90.8%	82.2%
P	84.8% ± 0.65%	92.2%	81.0%
SW	85.4% ± 0.64%	90.9%	83.6%
SWL	85.3% ± 0.64%	91.1%	83.2%
PW	85.9% ± 0.63%	92.2%	83.1%
PWL	86.2% ± 0.62%	92.1%	83.7%

Table 3: Summary of the experimental results on the PP attachment problem for various kernel parameters.

Results of the experiments are reported in Tab. 3 for various kernels parameters: S or P denote if the sum or product of the kernels between words are used, W denotes that WordNet semantic information is added (otherwise the kernel between two words is just the exact match kernel) and L denotes that lexicographer files identifiers are used. An additional gaussian kernel is used on top of K_{pp}. The C and γ parameters are selected using an independent validation set. For each setting, accuracy, precision and recall values on the test data are reported, along with the standard deviation of the estimated binomial distribution of errors. The results demonstrate that semantic information can help in resolving PP ambiguities. A small difference exists between taking the product instead of the sum of word kernels, and an additional increase in the amount of information available to the learner is given by the use of lexicographer files identifiers.

6 Using declarative knowledge for NLP kernel integration

Data objects in NLP often require complex representations; suffice it to say that a sentence is naturally represented as a variable length sequence of word tokens and that shallow/deep parsers are reliably used to enrich these representations with links between words to form parse trees. Finally, additional complexity can be introduced by including semantic information. Various facets of this richness of representations have been extensively investigated, including the expressiveness of various grammar formalisms, the exploitation of lexical representation (e.g. verb subcategorization, semantic tagging), and the use of machine readable sources of generic or specialized knowledge (dictionaries, thesauri, domain specific ontologies). All these efforts are capable to address language specific sub-problems but their integration into a coherent framework is a difficult feat.

Recent ideas for constructing kernel functions starting from logical representations may offer an appealing solution. Gärtner et al. (2002) have proposed a framework for defining kernels on higher-order logic individuals. Cumby and Roth (2003) used description logics to represent knowledge jointly with propositionalization for defining a kernel function. Frasconi et al. (2004) proposed kernels for handling supervised learning in a setting similar to that of inductive logic programming where data is represented as a collection of facts and background knowledge by a declarative program in first-order logic. In this section, we briefly review this approach and suggest a possible way of exploiting it for the integration of different sources of knowledge that may be available in NLP.

6.1 Declarative Kernels

The definition of decomposition kernels as reported in Section 2 is very general and covers almost all kernels for discrete structured data developed in the literature so far. Different kernels are designed by defining the relation decomposing an example into its "parts", and specifying kernels for individual parts. In (Frasconi et al., 2004) we proposed a systematic approach to such design, consisting in formally defining a relation by the set of axioms it must satisfy. We relied on *mereotopology* (Varzi, 1996) (i.e. the theory of parts and places) in order to give a formal definition of the intuitive concepts of parthood and connection. The formalization of mereotopological relations allows to automatically deduce instances of such relations on the data, by exploiting the background knowledge which is typically available on structured domains. In (Frasconi et al., 2004) we introduced *declarative kernels* (DK) as a set of kernels working on mereotopological relations, such as that of proper parthood ($\prec_P$) or more complex relations based on connected parts. A typed syntax for objects was introduced in order to provide additional flexibility in defining kernels on instances of the given relation. A basic kernel on parts K_P was defined as follows:

$$K_P(x,x') = \sum_{\substack{s \prec_P x \\ s' \prec_P x'}} \delta_T(s,s')\big(\kappa(s,s') + K_P(s,s')\big) \qquad (10)$$

where δ_T matches objects of the same type and κ is a kernel over object attributes.

Declarative kernels were tested in (Frasconi et al., 2004) on a number of domains with promising results, including a biomedical information extraction task (Goadrich et al., 2004) aimed at detecting protein-localization relationships within Medline abstracts. A DK on parts as the one defined in Eq. (10) outperformed state-of-the-art ILP-based systems Aleph and Gleaner (Goadrich et al., 2004) in such information extraction task, and required about three orders of magnitude less training time.

6.2 Weighted Decomposition Declarative Kernels

Declarative kernels can be combined with WDK in a rather straightforward way, thus taking the advantages of both methods. A simple approach is that of using proper parthood in place of selectors, and topology to recover the context of each proper part. A weighted decomposition declarative kernel (WD^{2}K) of this kind would be defined as in Eq. (10) simply adding to the attribute kernel κ a context kernel that compares the surrounding of a pair of objects—as defined by the topology relation—using some aggregate kernel such as PPK or HIK (see Section 3). Note that such definition extends WDK by adding recursion to the concept of comparison by selector, and DK by adding contexts to the kernel between parts. Multiple contexts can be easily introduced by employing different notions of topology, provided they are consistent with mereotopological axioms. As an example, if objects are words in a textual document, we can define l-connection as the relation for which two words are l-connected if there are consequential within the text with at most l words in between, and obtain growingly large contexts by increasing l. Moreover, an extended representation of words, as the one employing WordNet semantic information, could be easily plugged in by including a kernel for synsets such as that in Section 5.1 into the kernel κ on word attributes. Additional relations could be easily formalized in order to exploit specific linguisitc knowledge: a causal relation would allow to distinguish between preceding and following context so to take into consideration word order; an underlap relation, associating two objects being parts of the same super-object (i.e. pre-terminals dominated by the same non-terminal node), would be able to express commanding notions.

The promising results obtained with declarative kernels (where only very simple lexical information was used) together with the declarative ease to integrate arbitrary kernels on specific parts are all encouraging signs that boost our confidence in this line of research.

References

Roberto Basili, Marco Cammisa, and Alessandro Moschitti. 2005. Effective use of wordnet semantics via kernel-based learning. In *9th Conference on Computational Natural Language Learning*, Ann Arbor(MI), USA.

S. Ben-David, N. Eiron, and H. U. Simon. 2002. Limitations of learning via embeddings in euclidean half spaces. *J. of Mach. Learning Research*, 3:441–461.

M. Collins and N. Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of the Fortieth Annual Meeting on Association for Computational Linguistics*, pages 263–270, Philadelphia, PA, USA.

C. Cortes, P. Haffner, and M. Mohri. 2004. Rational kernels: Theory and algorithms. *J. of Machine Learning Research*, 5:1035–1062.

A. Culotta and J. Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 423–429.

C. M. Cumby and D. Roth. 2003. On kernel methods for relational learning. In *Proc. Int. Conference on Machine Learning (ICML'03)*, pages 107–114, Washington, DC, USA.

H. Daumé III and D. Marcu. 2005. Learning as search optimization: Approximate large margin methods for structured prediction. In *International Conference on Machine Learning (ICML)*, pages 169–176, Bonn, Germany.

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

P. Frasconi, S. Muggleton, H. Lodhi, and A. Passerini. 2004. Declarative kernels. Technical Report RT 2/2004, Università di Firenze.

T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola. 2002a. Multi-instance kernels. In C. Sammut and A. Hoffmann, editors, *Proceedings of the 19th International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann.

T. Gärtner, J.W. Lloyd, and P.A. Flach. 2002b. Kernels for structured data. In S. Matwin and C. Sammut, editors, *Proceedings of the 12th International Conference on Inductive Logic Programming*, volume 2583 of *Lecture Notes in Artificial Intelligence*, pages 66–83. Springer-Verlag.

T. Gärtner. 2003. A survey of kernels for structured data. *SIGKDD Explorations Newsletter*, 5(1):49–58.

M. Goadrich, L. Oliphant, and J. W. Shavlik. 2004. Learning ensembles of first-order clauses for recall-precision curves: A case study in biomedical information extraction. In *Proc. 14th Int. Conf. on Inductive Logic Programming, ILP '04*, pages 98–115.

D. Haussler. 1999. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, University of California, Santa Cruz.

T. Horváth, T. Gärtner, and S. Wrobel. 2004. Cyclic pattern kernels for predictive graph mining. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 158–167. ACM Press.

T. Jebara, R. Kondor, and A. Howard. 2004. Probability product kernels. *J. Mach. Learn. Res.*, 5:819–844.

H. Kashima and T. Koyanagi. 2002. Kernels for Semi–Structured Data. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 291–298.

H. Kashima, K. Tsuda, and A. Inokuchi. 2003. Marginalized kernels between labeled graphs. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 321–328, Washington, DC, USA.

C. S. Leslie, E. Eskin, and W. S. Noble. 2002. The spectrum kernel: A string kernel for SVM protein classification. In *Pacific Symposium on Biocomputing*, pages 566–575.

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444.

S. Menchetti, F. Costa, and P. Frasconi. 2005. Weighted decomposition kernels. In *Proceedings of the Twenty-second International Conference on Machine Learning*, pages 585–592, Bonn, Germany.

Alessandro Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *42-th Conference on Association for Computational Linguistic*, Barcelona, Spain.

F. Odone, A. Barla, and A. Verri. 2005. Building kernels from binary strings for image matching. *IEEE Transactions on Image Processing*, 14(2):169–180.

A Ratnaparkhi, J. Reynar, and S. Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 250–255, Plainsboro, NJ.

C. Saunders, H. Tschach, and J. Shawe-Taylor. 2002. Syllables and other string kernel extensions. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 530–537.

B. Schölkopf, J. Weston, E. Eskin, C. S. Leslie, and W. S. Noble. 2002. A kernel approach for learning from almost orthogonal patterns. In *Proc. of ECML'02*, pages 511–528.

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

G. Siolas and F. d'Alche Buc. 2000. Support vector machines based on a semantic kernel for text categorization. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, volume 5, pages 205 – 209.

A.J. Smola and R. Kondor. 2003. Kernels and regularization on graphs. In B. Schölkopf and M.K. Warmuth, editors, *16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003*, volume 2777 of *Lecture Notes in Computer Science*, pages 144–158. Springer.

J Stetina and M Nagao. 1997. Corpus based pp attachment ambiguity resolution with a semantic dictionary. In *Proceedings of the Fifth Workshop on Very Large Corpora*, pages 66–80, Beijing, China.

J. Suzuki, H. Isozaki, and E. Maeda. 2004. Convolution kernels with feature selection for natural language processing tasks. In *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 119–126.

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proc. 21st Int. Conf. on Machine Learning*, pages 823–830, Banff, Alberta, Canada.

A.C. Varzi. 1996. Parts, wholes, and part-whole relations: the prospects of mereotopology. *Data and Knowledge Engineering*, 20:259–286.

D. Zelenko, C. Aone, and A. Richardella. 2003. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.