Semi-Supervised Abstraction-Augmented String Kernel for Multi-Level Bio-Relation Extraction

Pavel Kuksa[1], Yanjun Qi[2], Bing Bai[2], Ronan Collobert[2], Jason Weston[3], Vladimir Pavlovic[1], and Xia Ning[4]

1 Department of Computer Science, Rutgers University, USA
2 NEC Labs America, Princeton, USA
3 Google Research, New York City, USA
4 Computer Science Department, University of Minnesota, USA

Abstract. Bio-relation extraction (bRE), an important goal in bio-text mining, involves subtasks identifying relationships between bio-entities in text at multiple levels, e.g., at the article, sentence or relation level. A key limitation of current bRE systems is that they are restricted by the availability of annotated corpora. In this work we introduce a semi-supervised approach that can tackle multi-level bRE via string comparisons with mismatches in the string kernel framework. Our string kernel implements an abstraction step, which groups similar words to generate more abstract entities, which can be learnt with unlabeled data. Specifically, two unsupervised models are proposed to capture contextual (local or global) semantic similarities between words from a large unannotated corpus. This Abstraction-augmented String Kernel (ASK) allows for better generalization of patterns learned from annotated data and provides a unified framework for solving bRE with multiple degrees of detail. ASK shows effective improvements over classic string kernels on four datasets and achieves state-of-the-art bRE performance without the need for complex linguistic features.

Keywords: Semi-supervised string kernel, Relation extraction, Sequence classification, Learning with auxiliary information

1 Introduction

The task of relation extraction from text is important in biomedical domains, since most scientific discoveries describe biological relationships between bio-entities and are communicated through publications or reports. A range of text mining and NLP strategies have been proposed to convert natural language in the biomedical literature into formal computer representations to facilitate sophisticated biomedical literature access [14]. However, the lack of annotated data and the complex nature of biomedical discoveries have limited automatic literature mining from having large impact.

In this paper, we consider "bio-relation extraction" tasks, i.e. tasks that aim to discover biomedical relationships of interest reported in the literature through identifying the textual triggers with different levels of detail in the text [14]. Specifically we cover three tasks in our experiments associated with

Table 1. Examples of sentence-level task and relation-level task.

Task 2: Sentence-level PPI extraction	
Negative	TH, AADC and GCH were effectively co-expressed in transduced cells with three separate AAV vectors.
Positive	This study demonstrates that IL - 8 recognizes and activates CXCR1, CXCR2, and the Duffy antigen by distinct mechanisms .
Task 3: Relation-level PPI extraction	
Input Sentence	The protein product of c-cbl proto-oncogene is known to interact with several proteins, including Grb2, Crk, and PI3 kinase, and is known to regulate signaling...
Output Interacting Pairs	(c-cbl, Grb2), (c-cbl, Crk), (c-cbl, PI3).

one important biological relation: protein-protein-interaction (PPI). In order to identify PPI events, the tasks aim to: (1) retrieve PubMed abstracts describing PPIs; (2) classify text sentences as PPI relevant or not relevant; (3) when protein entities have been recognized in the sentence, extract which protein-protein pairs having interaction relationship, i.e. pairwise PPI relations from the sentence. Table 2 gives examples of the second and third tasks. Examples of the first task are long text paragraphs and are omitted due to space limitations.

There exist very few annotated training datasets for all three tasks above. For bRE tasks at article-level, researchers [14] handled them as text categorization problems and support vector machines were shown to give good results with careful pre-processing, stemming, POS and named-entity tagging, and voting. For bRE tasks at the relation level, most systems in the literature are rule-based, cooccurrence-based or hybrid approaches (survey in [29]). Recently several researchers proposed the all-paths graph kernel [1], or an ensemble of multiple kernels and parsers [21], which were reported to yield good results.

Generally speaking, these tasks are all important instances of information extraction problems where entities are protein names and relationships are protein-protein interactions. Early approaches for the general "relation extraction" problem in natural languages are based on patterns [23], usually expressed as regular expressions for words with wildcards. Later researchers proposed kernels for dependency trees [7] or extended the kernel with richer structural features [23]. Considering the complexity of generating dependency trees from parsers, we try to avoid this step in our approach. Also bRE systems at article/long-text levels need to handle very long word sequences, which are problematic for previous tree/graph kernels to handle.

Here we propose to detect and extract relations from biomedical literature using string kernels with semi-supervised extensions, named Abstraction-augmented String Kernels (ASK). A novel semi-supervised "abstraction" augmentation strategy is applied on a string kernel to leverage supervised event extraction with unlabeled data. The "abstraction" approach includes two stages: (1) Two unsupervised auxiliary tasks are proposed to learn accurate word representations from contextual semantic similarity of words in biomedical liter-

ature, with one task focusing on short local neighborhoods (local ASK), and the other using long paragraphs as word context (global ASK). (2) Words are grouped to generate more abstract entities according to their learned representations. On benchmark PPI extraction data sets targeting three text levels, the proposed kernel achieves state-of-the-art performance and improves over classic string kernels.

Furthermore, we want to point out that ASK is a general sequence modeling approach and not tied to the multi-level bRE applications. We show this generality by extending ASK to a benchmark protein sequence classification task (the 4th dataset), and get improved performances over all tested supervised and semi-supervised string kernel baselines.

2 String Kernels

All of our targeted bRE tasks can be treated as problems of classifying sequences of words into certain types related to the relation of interest (i.e., PPI). For example, in bRE tasks at the article-level, we classify input articles or long paragraphs as PPI-relevant (positive) or not (negative). For the bRE task at the sentence-level, we classify sentence into PPI-related or not, which again is a string classification problem.

Various methods have been proposed to solve the string classification problem, including generative (e.g., HMMs) or discriminative approaches. Among the discriminative approaches, string kernel-based machine learning methods provide some of the most accurate results [27, 19, 16, 28].

The key idea of basic string kernels is to apply a mapping $\phi(\cdot)$ to map text strings of variable length into a vectorial feature space of fixed length. In this space a standard classifier such as a support vector machine (SVM) can then be applied. As SVMs require only inner products between examples in the feature space, rather than the feature vectors themselves, one can define a *string kernel* which *implicitly* computes an inner product in the feature space:

$$K(x, y) = \langle \phi(x), \phi(y) \rangle, \tag{1}$$

where $x, y \in S$, S is the set of all sequences composed of elements which take on a finite set of possible values, e.g., sequences of words in our case, and $\phi : S \rightarrow \mathbb{R}^m$ is a feature mapping from a word sequence (text) to a m-dim. feature vector.

Feature extraction and feature representation play key roles in the effectiveness of sequence analysis since text sequences cannot be readily described as feature vectors. Traditional text categorization methods use feature vectors indexed by all possible words (e.g., bag of words [25]) in a certain dictionary (vocabulary $\mathcal{D}$) to represent text documents, which can be seen as a simple form of string kernel. This "bag of words" strategy treats documents as an unordered set of features (words), where critical word ordering information is not preserved. To take word ordering into account, documents can be considered as bags of short sequences of words with feature vectors corresponding to all possible word n-grams (n adjacent words from vocabulary $\mathcal{D}$). With this representation, the

Table 2. Subsequences considered for string matching in different kernels.

Type	Parameters	Subsequences to Consider
Spectrum Kernel	$k = 3$	(SM binds RNA), (binds RNA in), (RNA in vitro), ...
Mismatch Kernel	$k = 3, m = 1$	(X binds RNA), (SM X RNA), (SM binds X), (X RNA in), (binds X in), (binds RNA X), ...
Gapped Kernel	$k = 3, m = 1$	(SM [ ] RNA in), (binds RNA in [ ]), (binds [ ] in vitro), ...

high similarity between two text documents means they have many n-grams in common. One can then define a corresponding string kernel as follows,

$$K(x, y) = \sum_{\gamma \in \Gamma} c_x(\gamma) \cdot c_y(\gamma), \tag{2}$$

where γ is a n-gram, Γ is the set of all possible n-grams, and $c_x(\gamma)$ is the number of occurrences (with normalization) of n-gram γ in a text string x. This is also called the spectrum kernel in the literature [18]. More general, the so-called substring kernels [27] measure similarity between sequences based on common co-occurrence of exact sub-patterns (e.g., substrings).

Inexact comparison, which is critical for effective matching (similarity evaluation) between text documents due to naturally occurring word substitutions, insertions, or deletions, is typically achieved by using different families of mismatch [19]. The mismatch kernel considers word (or character) n-gram counts with inexact matching of word (or character) n-grams. The gapped kernel calculates dot-product of (non-contiguous) word (or character) n-gram counts with gaps allowed between words. That is we revise $c_x(\gamma)$ as the number of subsequences matching the n-gram γ with up to k gaps. For example, as shown in Table 2, when calculating counts of trigram in a given sentence "SM binds RNA in vitro ..." , three string kernels we tried in our experiments consider different subsequences into the counts. As can be seen from examples, string kernels can capture relationship patterns using mixtures of words (n-grams with gaps or mismatch) as features.

String kernel implementations in practice typically require efficient methods for dot-product computation without explicitly constructing potentially very high-dimensional feature vectors. A number of algorithmic approaches have been proposed [27, 24, 17] for efficient string kernel computation and we adopt a sufficient statistic strategy from [16] for fast calculation of mismatch and gapped kernels. It provides a new family of *linear* time string kernel computation that scale well with large alphabet size and input length, e.g., word vocabulary in our context.

3 ASK: Abstraction-augmented String Kernel

Currently there exist very few annotated training data for the tasks of bio-relation extractions. For example, the largest (to the best of our knowledge)

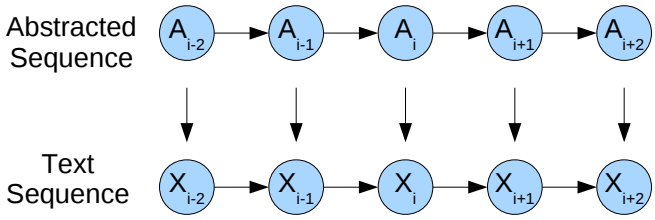

Fig. 1. Semi-supervised Abstraction-Augmented String Kernel. Both text sequence X and learned abstracted sequence A are used jointly.

publicly available training set for identifying "PPI relations" from PubMed abstracts includes only about four thousands annotated examples. This small set of training data could hardly cover most of the words in the vocabulary (about 2 million words in PubMed, which is the central collection of biomedical papers). On the other hand, PubMed stores more than 17 million citations (papers/reports), and provides free downloads of all abstracts (with over ∼1.3G tokens after preprocessing). Thus our goal is to use a large unlabeled corpus to boost the performance of string kernels where only a small number of labeled examples are provided for sequence classification.

We describe a new semi-supervised string kernel, called "Abstraction-augmented String Kernel" (ASK). The key term "abstraction" describes an operation of grouping similar words to generate more abstract entities. We also refer to the resulting abstract entities as "abstraction". ASK is accomplished in two steps: (i) learning word abstractions with unsupervised embedding and clustering (Figure 2); (ii) constructing a string kernel on both words and word abstractions (Figure 1).

3.1 Word Abstraction with Embedding

ASK relies on the key observation that individual words carry significant semantic information in natural language text. We learn a mapping of each word to a vector of real values (called an "embedding" in the following) which describes this word's semantic meaning. Figure 2 illustrates this mapping step with an exemplar sentence. Two types of unsupervised auxiliary tasks are exploited to learn embedded feature representations from unlabeled text, which aim to capture:

- Local semantic patterns: an unsupervised model is trained to capture words' semantic meanings in short text segments (e.g. text windows of 7 words).
- Global semantic distribution: an unsupervised model is trained to capture words' semantic patterns in long text sequences (e.g. long paragraphs or full documents).

Local Word Embedding (Local ASK) It can be observed that in most natural language text, semantically similar words can usually be exchanged with no impact on the sentence's basic meaning. For example, in a sentence like "EGFR interacts with an inhibitor" one can replace "interacts" with "binds" with no change in the sentence labeling. With this motivation, traditional language models estimate the probability of the next word being w in a language sequence. In

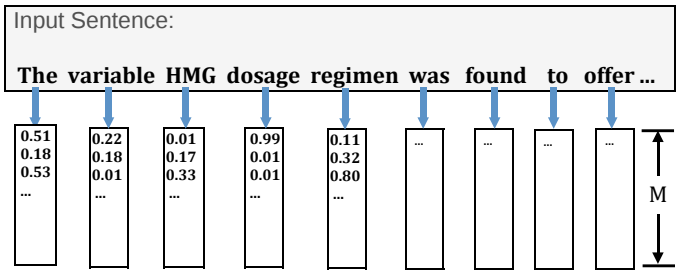

Fig. 2. The word embedding step maps each word in an input sentence to a vector of real values (with dimension M) by learning from a large unlabeled corpus.

a related task, [6] proposed a different type of "language modeling"(LM) which learns to embed normal English words into a M dimensional feature space by utilizing unlabeled sentences with an unsupervised auxiliary task. We adapt this approach to bio-literature texts and train the language model on unlabeled sentences in PUBMED abstracts.

We construct an auxiliary task which learns to predict whether a given text sequence (short word window) exists naturally in biomedical literature, or not. The real text fragments are labeled as positive examples, and negative text fragments are generated by random word substitution (in this paper we substitute the middle word by a random word). That is, LM tries to recognize if the word in the middle of the input window is related to its context or not. Note, the end goal is not the solution to the classification task itself, but the embedding of words into an M-dimensional space that are the parameters of the model. These will be used to effectively *learn the abstraction* for ASK.

Following [6], a Neural Network (NN) architecture is used for this LM embedding learning. With a sliding window approach, values of words in the current window are concatenated and fed into subsequent layers which are classical neural network (NN) layers (with one hidden layer and another output layer, using sliding text windows of size 11). The word embeddings and parameters of the subsequent NN layers are all automatically trained by backpropagation. The model is trained with a ranking-type cost (with margin):

$$\sum_{s \in \mathcal{S}} \sum_{w \in \mathcal{D}} \max\left(0,\, 1 - f(s) + f(s^w)\right) , \tag{3}$$

where $\mathcal{S}$ is the set of possible local windows of text, $\mathcal{D}$ is the vocabulary of words, and $f(\cdot)$ represents the output of NN architecture and s^w is a text window where the middle word has been replaced by a random word w (negative window as mentioned above). These learned embeddings give good representations of words where we take advantage of the complete context of a word (before and after) to predict its relevance. The training is handled with stochastic gradient descent which samples the cost *online* w.r.t. (s, w).

Global Word Embedding (Global ASK) Since the local word embedding learns from very short text segments, it cannot capture similar words having long range relationships. Thus we propose a novel auxiliary task which aims to catch word semantics within longer text sequences, e.g., full documents.

We still represent each word as a vector in an M dimensional feature space as in Figure 2. To capture semantic patterns in longer texts, we try to model real articles in an unlabeled language corpus. Considering that words happen multiple times in documents, we represent each document as a weighted sum of its included words' embeddings,

$$g(d) = \sum_{w \in d} c_d(w) E(w) \tag{4}$$

where scalar $c_d(w)$ means the normalized tf-idf weight of word w on document d, and vector $E(w)$ is the M-dim embedded representation of word w which would be learned automatically through backpropagation. The M-dimensional feature vector $g(d)$ thus represents the semantic embedding of the current document d.

Similar to the LM, we try to force $g(\cdot)$ of two documents with similar meanings to have closer representations, and force two documents with different meanings to have dissimilar representations. For an unlabeled document set, we adopt the following procedure to generate a pseudo-supervised signals for training of this model. We split a document a into two sections: a_0 and a_1, and assume that (in natural language) the similarity between two sections a_0 and a_1 is larger than the similarity between a_i ($i \in \{0, 1\}$) and one section b_j ($j \in \{0, 1\}$) from another random document b: that is

$$f(g(a_0), g(a_1)) > f(g(a_i), g(b_j)) \tag{5}$$

where $f(\cdot)$ represents a similarity measure on the document representation $g(\cdot)$. $f(\cdot)$ is chosen as the cosine similarity in our experiments. Naturally the above assumption comes to minimize a margin ranking loss:

$$\sum_{(a,b) \in \mathcal{A}} \sum_{i,j=0,1} \max(0, 1 - f(g(a_i), g(a_{1-i})) + f(g(a_i), g(b_j))) \tag{6}$$

where $i \in \{0, 1\}$, $j \in \{0, 1\}$ and $\mathcal{A}$ represents all documents in the unlabeled set. We train $E(w)$ using stochastic gradient descent, where iteratively, one picks a random tuple from (a_i and b_j) and makes a gradient step for that tuple. The stochastic method scales well to our large unlabeled corpus and is easy to implement.

Abstraction using Vector Quantization As we mentioned, "abstraction" means grouping similar words to generate more abstract entities. Here we try to group words according to their embedded feature representations from either of the two embedding tasks described above. For a given word w, the auxiliary tasks learn to define a feature vector $E(w) \in \mathbb{R}^M$. Similar feature vectors $E(w)$ can indicate semantic closeness of the words. Grouping similar $E(w)$ into compact entities might give stronger indications of the target patterns. Simultaneously, this will also make the resulting kernel tractable to compute[1].

[1] One could avoid the VQ step by considering the direct kernel $k(x, y) = \sum_{i,j} \exp(-\gamma||E(x_i) - E(y_j))||)$ that measures the similarity of embeddings between all pairs of words between two documents, but this would be slow to compute.

Table 3. Example words mapped to the same "abstraction" as the query word (first column) according to two different embeddings. We can see that "local" embedding captures part-of-speech and "local" semantics, while "global" embedding found words semantically close in their long range topics across a document.

Query	**Local ASK**	**Global ASK**
protein	ligand, subunit, receptor, molecule	proteins, cosNUM, phosphoprotein, isoform
medical	surgical, dental, preventive, reconstructive	hospital, investigated, research, urology
interact	cooperate, compete, interfere, react	interacting, interacts, associate, member
immunoprecipitation	co-immunoprecipitation, EMSA, autoradiography, RT-PCR	coexpression, two-hybrid, phosphorylated, tbp

As a classical lossy data compression method in the field of signal processing, Vector quantization (VQ) [10] is utilized here to achieve the abstraction operation. The input vectors are quantized (clustered) into different groups via "prototype vectors". VQ summarizes the distribution of input vectors with their matched prototype vectors. The set of all *prototype vectors* is called the *codebook*. We use $\mathcal{C}$ to represent the codebook set which includes N prototype vectors, $\mathcal{C} = \{C_1, C_2, ..., C_N\}$.

Formally speaking, VQ tries to optimize (minimize) the following objective function, in order to find the codebook $\mathcal{C}$ and in order to best quantize each input vector into its matched prototype vector,

$$\sum_{i=1...|\mathcal{D}|} ||E(w_i) - C_n||^2, \ n \in \{1...N\} \tag{7}$$

where $E(w_i) \in \mathbb{R}^M$ is the embedding of word w_i. Hence, our basic VQ is essentially a k-means clustering approach.

For a given word w we call the index of the prototype vector C_j that is closest to $E(w)$ its *abstraction*.

According to the two different embeddings, Table 3 gives the lists of example words mapped to the same "abstraction" as the query word (first column). We can see that "local" embedding captures part-of-speech and "local" semantics, while "global" embedding found words semantically close in their long range topics across a document.

3.2 Semi-Supervised String Kernel

Unlike standard string kernels which use words directly from the input text, semi-supervised ASK combines word sequences with word abstractions (Figure 1). The word abstractions are learned to capture local and global semantic patterns of words (described in previous sections). As Table 3 shows, using learned embeddings to group words into abstractions could give stronger indications of the target pattern. For example, in local ASK, the word "protein" is grouped with terms like "ligand", "receptor", or "molecule". Clearly, this abstraction could

improve the string kernel matching since it provides a good summarization of the involved parties related to target event patterns.

We define the semi-supervised abstraction-augmented string kernel as follows

$$K(x,y) = \Big\langle \big(\phi(x), \phi'(a(x))\big), \big(\phi(y), \phi'(a(y))\big) \Big\rangle \tag{8}$$

where $(\phi(x), \phi'(a(x)))$ extends the basic n-gram representation $\phi(x)$ with the representation $\phi'(a(x))$. $\phi'(a(x))$ is a n-gram representation of the abstraction sequence, where

$$a(x) = (a(x_1), \ldots, a(x_{|x|})) = (A_1, \ldots, A_{|x|}) \tag{9}$$

$|x|$ means the length of the sequence and its i_{th} item is $A_i \in \{1...N\}$. The abstraction sequence $a(x)$ is learned through the embedding and abstraction steps.

The abstraction kernel exhibits a number of properties:

- It is a wrapper approach and can be used to extend *both supervised* and *semi-supervised* string kernels.
- It is very efficient as it has linear cost in the input length.
- It provides two unsupervised models for word-feature learning from unlabeled text.
- The baseline supervised or semi-supervised models can learn if the learned abstractions are relevant or not.
- It provides a unified framework for bRE at multiple levels where tasks have small training sets.
- It is quite general and not restricted to the biomedical text domain, since no domain specific knowledge is necessary for the training.
- It can incorporate other types of word similarities (e.g., obtained from classical latent semantic indexing [8]).

4 Related Work

4.1 Semi-supervised Learning

Supervised NLP techniques are restricted by the availability of labeled examples. Semi-supervised learning has become popular, since unlabeled language data is abundant. Many semi-supervised learning algorithms exist, including self-training, co-training, Transductive SVMs, graph-based regularization [30], entropy regularization [11] and EM with generative mixture models [22], see [5] for a review. Except self-training and co-training, most of these semi-supervised methods have scalability problems for large scale tasks.

Some other methods utilized auxiliary information from large unlabeled corpora for training sequence models (e.g., through multi-task learning). Ando and Zhang [2] proposed a method based on defining multiple tasks using unlabeled data that are multi-tasked with the task of interest, which they showed to perform very well on POS and NER tasks. Similarly, the language model strategy proposed in [6] is another type of auxiliary task. Both our local and global embedding methods belong to this semi-supervised category.

4.2 Semi-Supervised String kernel

For text categorization, the word sequence kernel proposed in [4] utilizes soft matching of words based on a certain similarity matrix used within the strin kernels. This similarity matrix could be derived from cooccurrence of words in unlabled text, i.e. adding semi-supervision to string kernel. Adding soft matching in the string kernel results qudratic complexity, though ASK does not add to complexity more than a linear cost to the input length (in practice we observed at most a factor of 1.5-2x slowdown compared to classic string kernels), while improving predictive performance significantly (Section "Results").

In terms of semi-supervised extensions of string kernels, another very simple method, called the "sequence neighborhood" kernel or "cluster" kernel has been employed [28] previously. This method replaces every example with a new representation obtained by averaging representations of the example's neighbors found in the unlabeled data using some standard sequence similarity measure. This kernel applies well in biological sequence analysis since relatively accurate measures exist (e.g., PSI-BLAST). Formally speaking, the *sequence neighborhood* kernels take advantage of the unlabeled data using the process of neighborhood induced regularization. But its application in most other domains (like text) is not straightforward since no accurate and standard measure of similarity exists.

4.3 Word Abstraction Based Models

Several previous works ([20]) tried to solve information extraction tasks with word clustering (abstraction). For example, Miller et al. [20] proposed to augment annotated training data with hierarchical word clusters that are automatically derived from a large unannotated corpus according to occurrence. Another group of closely related methods treat word clusters as hidden variables in their models. For instance, [12] proposed a conditional log-linear model, with hidden variables representing the assignment of atomic items to word clusters or word senses. The model learns to automatically make the cluster assignments based on a discriminative training criterion. Furthermore, researchers proposed to augment probabilistic models with abstractions in a hierarchical structure [26]. Our proposed ASK differs by building words similarity from two unsupervised models to capture auxiliary information implicit in large text corpus and employs VQ to build discrete word groups for string kernels.

5 Experimental Results

We now present experimental results for comparing ASk to classic string kernels and the state-of-art bRE results at multiple levels. Moreover to show generality, we extend ASK and apply it to a benchmark protein sequence classification dataset as the fourth experiment.

5.1 Three Benchmark bRE Data Sets

In our experiments, we explore three benchmark data sets related to PPI relation extractions. (1) The first one was provided from BioCreative II [13], a

Table 4. Size of datasets used in three "relation extraction" tasks

Dataset	**Labeled**	**Unlabeled**
BioCreativeII IAS Train	5495 (abstracts)1142559(tokens)	4.5M (abstracts)∼1.3G (tokens)
BioCreativeII IAS Test	677 (abstracts)143420 (tokens)	
AIMED Relation	4026 (sentences) 143774 (tokens)	4.5M (abstracts)∼1.3G (tokens)
AIMED Sentence	1730 (sentences)50675 (tokens)	4.5M (abstracts)∼1.3G (tokens)

competition in 2006 for the extraction of protein-protein interaction (PPI) annotations from the literature. The competition evaluated multiple teams' submissions against a manually curated "gold standard" carried out by expert database annotators. Multiple subtasks were tested and we choose one specific task called "IAS" which aims to classify PubMed abstracts, based on whether they are relevant to protein interaction annotation or not. (2) The second data set is the "AIMED PPI sentence classification" data set. Extraction of relevant text segments (sentences) containing reference to important biomedical relationships is one of the first steps in annotation pipelines of biomedical database curation. Focusing on PPI, this step could be accomplished through classification of text fragments (sentences) as either relevant (i.e. containing PPI relation) or not relevant (non-PPI sentences). Sentences with PPI relations in the AIMED dataset [3] are treated as positive examples, while all other sentences (without PPI) are negative examples. In this data set, protein names are not annoated. (3) The third data set is called "AIMED PPI Relation Extraction", which uses a benchmark set aiming to extract binary protein-protein interaction (PPI) pairs from bio-literature sentences [3]. An example of such extraction is listed in Table 2. In this set, the sentences have been annotated with protein names if any. To ensure generalization of the learned extraction model, protein names are replaced with PROT1, PROT2 or PROT, where PROT1 and PROT2 are the pair of interests. The PPI relation extraction task is treated as a binary classification, where protein pairs that are stated to interact are positive examples and other co-occurring pairs negative. This means, for each sentence, $\binom{n}{2}$ relation examples are generated, with n as the number of protein names in the sentence. We downloaded this corpus from [9].

We use over 4.5M PubMed abstracts from 1994 to 2009 as our unlabeled corpus for learning word abstractions. The size of the training/test/unlabeled sets is given in Table 4.

Baselines As each of these datasets has been used extensively, we will also compare our methods with the best reported results in the literature (see Table 5 and 7). In the following, we also compare global and local ASK with various other baselines string kernels, including fully-supervised and semi-supervised approaches.

Method We used the word n-grams as base features with ASK. Note we did not use any syntactic or linguistic features (e.g., no POS, chunk types, parse tree attributes, etc).

Table 5. Comparison with previous results and baselines on IAS task.

Method	Precision	Recall	F1	ROC	Accuracy
Baseline 1: BioCreativeII compet. (best)	70.31	**87.57**	78.00	81.94	75.33
Baseline 2: BioCreativeII compet. (rank-2)	75.07	81.07	77.95	84.71	77.10
Baseline 3: TF-IDF	66.83	82.84	73.98	79.22	70.90
Spectrum (n-gram) kernel	69.29	80.77	74.59	81.49	72.53
Mismatch kernel	69.02	83.73	75.67	81.70	73.12
Gapped kernel	67.84	85.50	75.65	82.01	72.53
Global ASK	73.59	84.91	78.85	84.96	77.25
Local ASK	**76.06**	84.62	**80.11**	**85.67**	**79.03**

For global ASK, we use PubMed abstracts to learn word embedding vectors using a vocabulary of the top 40K most frequent words in PubMed. These word representations are clustered to obtain word abstractions (1K prototypes). Similarly, local ASK learns word embeddings on text windows (11 words, with 50-dim. embedding) extracted from the PubMed abstracts. Word embeddings are again clustered to obtain 1K abstraction entities. We set parameters of the string kernels to typical values, with spectrum n-gram using $k = 1$ *to* 5, the maximum number of mismatches is set to $m = 1$ and the maximum number of gaps uses up to $g = 6$).

Metric The methods are evaluated using F1 score (including precision and recall) as well as ROC score. (1) For BioCreativeII IAS, evaluation is performed at the document level. (2) For two "AIMED" tasks, PPI extraction performance is measured at the sentence level for predicted/extracted interacting protein pairs using 10-fold cross-validation.

5.2 Task 1: PPI extract at article-level: IAS

The lower part of Table 5 summarizes results for the IAS task from Global and Local ASK to baseline methods (spectrum n-gram kernel, n-gram kernel with mismatches, and gapped n-gram kernel using different base feature sets (words only, stems, characters)). Both Local and Global ASK provide improvements over baseline n-gram based string kernels.

Using word and character n-gram features, the best performance obtained with global ASK (F1 78.85), and the best performance by local ASK (F1 80.11) are superior to the best performance reported in the BioCreativeII competition (F1 78.00), as well as baseline bag-of-words with TF-IDF weighting (F1 73.98) and the best supervised string kernel result in the competition (F1 77.17). Observed improvements are significant, e.g., local ASK (F1 80.11) performs better than the best string kernel (F1 77.17), with p-values 5.8e-3 (calculating with standard z-test).

Note that all the top systems in the competition used more extensive feature sets than ours, including protein names, interaction keywords, part of speech tags and/or parse trees, etc. Thus, in summary, ASK effectively improves interaction article retrieval and achieves state-of-the-art performance with only plain words

Table 6. AIMED PPI sentence classification task (F1 score). Both local ASK and global ASK improve over string kernel baselines.

Method	**Baseline**	**+Global ASK**	**+Local ASK**
Words	61.49	67.83	**69.46**
Words+Stems	65.94	67.99	**70.49**

Table 7. Comparison with previous results and baselines on AIMED relation-leve data

Method	**Precision**	**Recall**	**F1**	**ROC**	**Accuracy**
Baseline 1: Bag of words	41.39	62.46	49.75	74.58	70.22
Baseline 2: Transductive SVM [9]	59.59	60.68	59.96	-	-
Spectrum n-gram	58.35	62.77	60.42	83.06	80.57
Mismatch kernel	52.88	59.83	56.10	77.88	71.89
Gapped kernel	57.33	64.35	60.59	82.47	80.53
Global ASK	60.68	**69.08**	**64.54**	84.94	82.07
Local ASK	**61.18**	67.92	64.33	**85.27**	82.24

as features. We also note that using both local and global ASK together (multiple kernel) provides further improvements in performance compared to individual kernel results (e.g., we observe an increase in F1 score to 80.22)

5.3 Task 2: PPI extraction sentence level: AIMED PPI sentence

For the third benchmark task, "Classification of Protein Interaction Sentences", we summarize comparison results of both local and global ASK in Table 6.

The task here is to classify sentences as containing PPI relations or not. Both ASK models effectively improve over the traditional spectrum n-gram string kernels. For example, F1 70.49% from local ASK is significantly better than F1 65.94% from the best string kernel.

5.4 Task 3: PPI extraction relation-level: AIMED

Table 7 summarizes the comparison results between ASK to baseline bag-of-words and supervised string kernel baselines. Both local and global ASK show effective improvements over the word n-gram based string kernels. We find that the observed improvements are statistically significant with $p < 0.05$ for the case with the best performance (F1 64.54) achieved by global ASK. One state-of-the-art relation-level bRE system (as far as we know) is listed as "baseline 2" in Table 7, which was tested on the same AIMED dataset as we used. Clearly our approach (with 64.54 F-score) performs better than this baseline (59.96 F-score) while using only basic words. Moreover, this baseline system utilized many complex, expensive techniques such as, dependency parsers, to achieve good performance.

Furthermore as pointed out by [1], though the AIMED corpus has been applied in numerous evaluations for PPI relation extraction, the datasets used in different papers varied largely due to diverse postprocessing rules used to create the relation-level examples. For instance, the corpus used to test our ASK in

Table 7 was downloaded from [9] which contains 4026 examples with 951 as positive and 3075 as negatives. However, the AIMED corpus used in [1] includes more relation examples, i.e. 1000 positive relations and 4834 negative examples. The difference between the two reference sets make it impossible to compare our results in Table 7 to this state-of-the-art bRE system as claimed by [1] (with 56.4 F-score). Therefore we re-experiment ASK on this new AIMED relation corpus with both local and global ASK using the mismatch or spectrum kernel. Under the same (abstract-based) cross-validation splits from [1], our best performing case could achieve 54.7 F-score from local ASK on spectrum n-gram kernel with k from 1 to 5. We conclude that using only basic words ASK is comparable (slightly lower) to the bRE system from [1] where complex POS tree structures were used.

5.5 Task 4: Comparison on biological sequence task

As mentioned in the introduction, the proposed ASK method is general to any sequence modeling problem, and good for cases with few labeled examples and a large unlabeled corpus. In the following, we extend ASK to biological domain and compare it with semi-supervised and supervised string kernels . The related work Section pointed out that the "Cluster kernel" is the only realistic semi-supervised competitor we know so far proposed for string kernels. However it needs a similarity measure specific to "protein sequences", which is not applicable to most sequence mining tasks. Three benchmark datasets evaluated above are all within the scope of text mining, where the cluster kernel is not applicable. In this experiment, we compare ASK with the cluster kernel and other string kernels in the biological domain on the problem of structural classification from protein sequences.

Measuring the degree of structural homology between protein sequences (also known as remote protein homology prediction) is a fundamental and difficult problem in biomedical research. For this problem, we use a popular benchmark dataset for structural homology prediction (SCOP) that corresponds to 54 remote homology detection experiments [28, 17]. We test local ASK (with local embedding trained on a UNIPROT dataset, a collection of about 400,000 protein sequences) and compare with the supervised string kernels commonly used for the remote homology detection [19, 28, 15, 17]. Each amino acid is treated as a word in this case. As shown in Table 8, local ASK effectively improves the performance of the traditional string kernels. For example, the mean ROC50 score (commonly used metric for this task) improves from 41.92 to 46.68 in the case of the mismatch kernel. One reason for this may be the use of the abstracted alphabet (rather than using standard amino-acid letters) which effectively captures similarity between otherwise symbolically different amino-acids. We also observe that adding ASK on the semi-supervised cluster kernel approach [28] improves over the standard mismatch string kernel-based cluster kernel. For example, for the cluster kernel computed on the unlabeled subset ($\sim$ 4000 protein sequences) of the SCOP dataset, the cluster kernel with ASK achieves mean ROC50 70.14 compared to ROC50 67.91 using the cluster kernel alone.

Table 8. Mean ROC50 score on remote protein homology problem. Local ASK improves over string kernel baselines, both supervised and semi-supervised.

Method	**Baseline**	**+Local ASK**
Spectrum (n-gram)[18]	27.91	**33.06**
Mismatch [19]	41.92	**46.68**
Spatial sample kernel [15]	50.12	**52.75**
Semi-supervised Cluster kernel [28]	67.91	**70.14**

Furthermore the cluster kernel introduces new examples (sequences) and requires semi-supervision at testing time, while our unsupervised auxiliary tasks are feature learning methods, i.e. the learned features could be directly added to the existing feature set. From the experiments, it appears that the learned features from embedding models provide an orthogonal method for improving accuracy, e.g., these features could be combined with the cluster kernel to further improve its performance.

6 Conclusion

In this paper we propose to extract PPI relationships from sequences of biomedical text using a novel semi-supervised string kernel. The abstraction-augmented string kernel tries to improve supervised extractions with word abstractions learned from unlabeled data. Semi-supervision relies on two unsupervised auxiliary tasks that learn accurate word representations from contextual semantic similarity of words. On three bRE data sets, the proposed kernel matches state-of-the-art performance and improves over all string kernel baselines we tried without the need to get complex linguistic features. Moreover, we extend ASK to protein sequence analysis and on a classic benchmark dataset we found improved performance compared to all existing string kernels we tried.

Future work includes extension of ASK to more complex data types that have richer structures, such as graphs.

References

1. Airola, A., Pyysalo, S., Bjorne, J., Pahikkala, T., Ginter, F., Salakoski, T.: All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. BMC Bioinformatics 9(S11), S2 (2008)
2. Ando, R.K., Zhang, T.: A framework for learning predictive structures from multiple tasks and unlabeled data. J. of Machine Learning Research 6, 1817–1853 (2005)
3. Bunescu, R., Mooney, R.: Subsequence kernels for relation extraction. In: Weiss, Y., Schölkopf, B., Platt, J. (eds.) NIPS'06, pp. 171–178 (2006)
4. Cancedda, N., Gaussier, E., Goutte, C., Renders, J.M.: Word sequence kernels. J. Mach. Learn. Res. 3, 1059–1082 (2003)
5. Chapelle, O., Schölkopf, B., Zien, A. (eds.): Semi-Supervised Learning (Adaptive Computation and Machine Learning). MIT Press (2006)
6. Collobert, R., Weston, J.: A unified architecture for nlp: deep neural networks with multitask learning. In: ICML'08. pp. 160–167 (2008)

7. Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: ACL'04. p. 423 (2004)
8. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. JASIS 41(6), 391–407 (1990)
9. Erkan, G.: Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In: EMNLP-CoNLL'07. pp. 228–237 (2007)
10. Gersho, A., Gray, R.M.: Vector quantization and signal compression. Norwell, MA, USA (1991)
11. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: NIPS'05. pp. 529–536 (2005)
12. Koo, T., Collins, M.: Hidden-variable models for discriminative reranking. In: HLT '05. pp. 507–514 (2005)
13. Krallinger, M., Morgan, A., Smith, L., et al, Hirschman, L., Valencia, A.: Evaluation of text-mining systems for biology: overview of the second biocreative community challenge. Genome Biol 9(S2), S1 (2008)
14. Krallinger, M., Valencia, A., Hirschman, L.: Linking genes to literature: text mining, information extraction, and retrieval applications for biology. Genome Biol 9(S2), S8 (2008)
15. Kuksa, P., Huang, P.H., Pavlovic, V.: Fast protein homology and fold detection with sparse spatial sample kernels. In: ICPR 2008 (2008)
16. Kuksa, P., Huang, P.H., Pavlovic, V.: Scalable algorithms for string kernels with inexact matching. In: NIPS. pp. 881–888 (2008)
17. Leslie, C., Kuang, R.: Fast string kernels using inexact matching for protein sequences. J. Mach. Learn. Res. 5, 1435–1455 (2004)
18. Leslie, C.S., Eskin, E., Noble, W.S.: The spectrum kernel: A string kernel for SVM protein classification. In: PSB. pp. 566–575 (2002)
19. Leslie, C.S., Eskin, E., Weston, J., Noble, W.S.: Mismatch string kernels for SVM protein classification. In: NIPS. pp. 1417–1424 (2002)
20. Miller, S., Guinness, J., Zamanian, A.: Name tagging with word clusters and discriminative training. In: HLT-NAACL'04. pp. 337–342 (2004)
21. Miwa, Makoto, R.S.Y.M., Tsujii, J.: A rich feature vector for protein-protein interaction extraction from multiple corpora. In: EMNLP. pp. 121–130 (2009)
22. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Mach. Learn. 39(2-3), 103–134 (2000)
23. Reichartz, F., Korte, H., Paass, G.: Dependency tree kernels for relation extraction from natural language text. In: ECML. pp. 270–285 (2009)
24. Rousu, J., Shawe-Taylor, J.: Efficient computation of gapped substring kernels on large alphabets. J. Mach. Learn. Res. 6, 1323–1344 (2005)
25. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill Inc. (1986)
26. Segal, E., Koller, D., Ormoneit, D.: Probabilistic abstraction hierarchies. In: NIPS'01 (2001)
27. Vishwanathan, S., Smola, A.: Fast kernels for string and tree matching. vol. 15, pp. 569–576. MIT Press (2002)
28. Weston, J., Leslie, C., Ie, E., Zhou, D., Elisseeff, A., Noble, W.S.: Semi-supervised protein classification using cluster kernels. Bioinformatics 21(15), 3241–3247 (2005)
29. Zhou, D., He, Y.: Extracting interactions between proteins from the literature. J Biomed Inform 41(2), 393–407 (2008)
30. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: ICML'03. pp. 912–919 (2003)