JOURNAL OF
BIOMEDICAL SEMANTICS

RESEARCH Open Access

Protein interaction sentence detection using multiple semantic kernels

Tamara Polajnar[1]*, Theodoros Damoulas[2]* and Mark Girolami[3]

* Correspondence: tamara@dcs.gla.ac.uk; damoulas@cs.cornell.edu
[1]School of Computing Science, University of Glasgow, Glasgow, UK
[2]Department of Computer Science, Cornell University, 14850, Ithaca, NY, US
Full list of author information is available at the end of the article

Abstract

Background: Detection of sentences that describe protein-protein interactions (PPIs) in biomedical publications is a challenging and unresolved pattern recognition problem. Many state-of-the-art approaches for this task employ kernel classification methods, in particular support vector machines (SVMs). In this work we propose a novel data integration approach that utilises semantic kernels and a kernel classification method that is a probabilistic analogue to SVMs. Semantic kernels are created from statistical information gathered from large amounts of unlabelled text using lexical semantic models. Several semantic kernels are then fused into an overall composite classification space. In this initial study, we use simple features in order to examine whether the use of combinations of kernels constructed using word-based semantic models can improve PPI sentence detection.

Results: We show that combinations of semantic kernels lead to statistically significant improvements in recognition rates and receiver operating characteristic (ROC) scores over the plain Gaussian kernel, when applied to a well-known labelled collection of abstracts. The proposed kernel composition method also allows us to automatically infer the most discriminative kernels.

Conclusions: The results from this paper indicate that using semantic information from unlabelled text, and combinations of such information, can be valuable for classification of short texts such as PPI sentences. This study, however, is only a first step in evaluation of semantic kernels and probabilistic multiple kernel learning in the context of PPI detection. The method described herein is modular, and can be applied with a variety of feature types, kernels, and semantic models, in order to facilitate full extraction of interacting proteins.

Background

Proteins are the principal engine enabling chemical reactions in a cell, and, as such, are of great interest to biologists studying life on the molecular level. Part of the proteins' functionality depends on their interactions with each other. Information about these interactions is paramount to the understanding of pathologies, diseases, and treatments. The principal observations of interactions are made through biological experiments [1], whose results are reported in peer-reviewed biomedical journal articles. Protein-protein interactions (PPIs) are then found by researchers through various search engines indexing these specific articles. In text, a PPI is a relation between two protein entities linked by an action descriptor, which is usually either a verb, or a present (*-ing*) or past (*-ed*) participial adjective (e.g. *activate*, *activating*, *activated*). A relationship is difficult to describe using a query; therefore, current state-of-the-art search engines are not well suited for this task. In addition, *ad hoc* query-based searches are more appropriate for temporary information needs, not persistent ones [2]. For research tasks such as pathway construction or population of PPI databases such as KEGG [3], MIPS [4], or BIND [5], PPI extraction becomes a continuous process. Consequently, PPI detection and extraction have become one of the primary goals of biomedical text mining (TM) [6]. The aim is to develop applications that will enable habitual PPI searchers to find interactions without having to specify pairs of proteins or manually scan large amounts of text.

Automatic protein interaction detection can be useful in several different scenarios. There are, therefore, many different approaches for information extraction in the biomedical text. Some applications are geared towards helping with automatic population of interaction databases [7,8], while others aim to support a wide variety of users by bridging the gap between the search engines and highly customised relation extraction software [9-12]. Different approaches to PPI detection can be roughly categorised into pattern-based, information retrieval-based (IR-based), and classification-based [7,13-15].

Pattern-based systems consist of hand-coded or automatically induced templates derived from sample interaction sentences. The templates, which are sometimes scored for quality, are used to scan text and retrieve any matches. These patterns are usually unable to cover the wide variety of ways with which the interactions can be described in text. For this reason, these methods usually have high precision and lower recall. It is often offered as an argument that experimentally validated relations will be reported several times, thus affording more chance for the interaction to be retrieved [16,17]. Conversely, this approach may only retrieve well known interactions, and as such not be very helpful to a researcher looking for novel interactions in a field that she is familiar with.

On the other side of the spectrum are the methods that consider any co-occurrence of two proteins in a sentence as a possible interaction. This assumption leads to a large number of retrieved interactions, unfortunately with a very low precision rate. A favourite approach of initial systems aiming to construct interaction networks on the fly from user queries, it is an efficient way of allowing the user to browse potential interactions [9,10,18]. More advanced IR-based approaches incorporate interaction detection into the indexing process [11,12]. This allows for fast retrieval of highly detailed information. However, for new types of interactions or entities to be included, the entire collection needs to be re-indexed.

Finally, there are the (mainly supervised) classification-based methods [6,7,13,19-21]. These methods require samples of sentences that are, at the very least, annotated for relevance if not for the full interactions. On the other hand, they are fully automatic, apart from the labelling process. The availability of the standard data, such as AImed [20] and the LLL [22], has allowed for faster development and testing of new algorithms, as well as for comparison across different approaches [6,21,23,24].

What most of these systems have in common is the attempt to fully extract the interaction triples. In this paper, we step back to address a paired down problem: identification of sections of text that describe PPIs (in particular, sentences). In essence, the approach is similar to the latest classification-based methods, in that it employs state-of-the-art kernel classification. On the other hand, it uses bag-of-words [25] representation, which is easier to extract than the parser-based features, that are required for full triple extraction.

Using deep linguistic features increases the complexity of the approach by introducing performance variation with different choice of parser and kernel [26,27]. Although analogous systems have been developed for other domains, such as news, biomedical texts offer particular challenges that need to be addressed with tailored tools [6,28]. Even the detection of the protein names is a difficult problem, because of the high degree of synonymy, polysemy, orthographic variation, and novelty due to protein discovery [29-33]. Protein name recognition is not a necessary step for detection of areas of text that describe interactions [7,34], but for more detailed extraction it is essential [13-15,18,20,35,36].

The approach described in this paper consists of several components that themselves contain parameters that influence the performance of the method. Thus, to study the novelty of this approach we eliminate, as much as possible, reliance on further language processing algorithms. Consequently, by examining a simpler task, we produce results that are not directly comparable with the kernel-based PPI extraction methods described in [26], but are comparable to the baseline results described in [34,37]. However, the approach described here is modular, and can be augmented for use with methods that rely on deep linguistic features.

This paper introduces a method that improves the detection of sentences describing PPIs in biomedical texts. Classification-based methods are usually trained on data labelled by experts. The technique described herein is envisioned as a component of a trainable filtering system, which could be placed on top of a keyword search and could effectively learn from simple annotations provided by a user. For example, a user could indicate whether a sentence describes a PPI or not. Such annotation schemes would be less onerous than ones that require users to label each protein participating in an interaction, and perhaps any other words indicating their relationship. While any pattern recognition or discriminant analysis method could be used for this purpose, the main contribution of this paper is a novel method that enhances the effectiveness of learning from the labelled examples by incorporating semantic information from unlabelled data; and thus reducing the burden on the user.

Methods

Identification of interactions requires significant biological knowledge. In addition, annotation may also require grammatical expertise, depending on whether entities, interaction identifiers, or even sentence parse trees are considered. While quality labelled data is difficult to obtain in large quantities, unlabelled data is plentiful and freely available in the form of MEDLINE abstracts and full-text open access publications. Semi-supervised learning (SSL) [38,39] is a way to leverage the models trained on labelled data with large amounts of unlabelled data.

A novel approach to semi-supervised learning, where information collected from relevant large datasets, in an unsupervised manner, is incorporated directly into the training kernel was introduced in [34]. The unlabelled corpus is transformed into a matrix of term similarities, which is then projected onto the document vectors causing a rescaling of the labelled training data. In this paper, we extend this method further. Different semantic models can be used to calculate term similarities, each producing slightly different results. In order to take best advantage of the given semantic information, we combine these kernels using probabilistic Multiple Kernel Learning (pMKL).

pMKL is a method that, in single-kernel mode, produces similar results to Gaussian processes (GPs), which are comparable to the popularly used support vector machines (SVMs) [40]. Kernel combinations can be employed using any kernel method, as [14] do with SVMs; however, pMKL is also capable of estimating the best weighted combination of kernels. Whilst in this paper we use combinations of semantic kernels, multiple kernel algorithms can be used to combine various kernels of different feature-types in order to take advantage of several views of a single data set [26].

The rest of this section describes the components that are used in this method: the kernel learning algorithm and the semantic models. Firstly, the general description is given of how the components fit together, then a brief introduction to this particular kernel learning algorithm is provided. This is followed by an introduction to semantic models and then more detailed descriptions of the two models that are used here. This section concludes with a description of the experimental setup and is followed by a section that discusses the results of these experiments.

Semantic kernel construction and combination

The proposed method combines labelled and unlabelled data (semi-supervised learning), by integrating semantic information from unsupervised lexical semantic models trained on a larger corpus, such as the MEDLINE abstracts contained in the GENIA corpus [41]. It is described graphically in Figure 1.

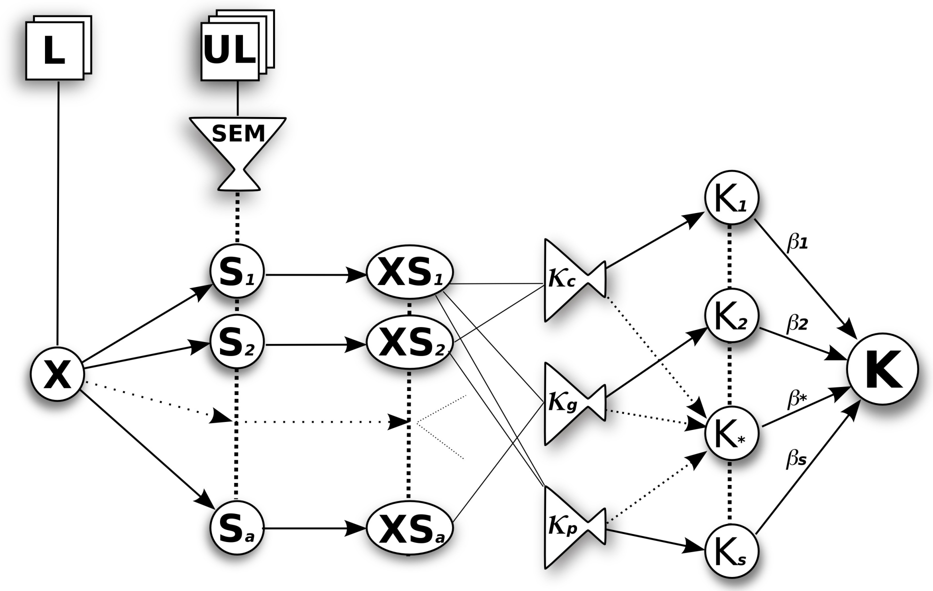


Figure 1 The overview of the method. The training data (**X**) comes from the labelled corpus (**L**), while the unlabelled data (**UL**) is transformed using semantic models (**SEM**) to produce smoothing matrices (**S**). The training data is then projected into the semantic subspace (**XS**) and passed into one or more of the available kernel functions. We use cosine (κ_c), Gaussian (κ_g), and polynomial (κ_p) kernels. We combine the resulting kernels with a weighting $\boldsymbol{\beta}_s$ into a single combined kernel (**K**).

Labelled training data is used for the probabilistic Multiple Kernel Learning (pMKL) classifier training and testing. The training data is represented as a $M \times N$ matrix $\mathbf{X}$ (where there are M documents and N features), and an $M \times 1$ vector of training labels.

We then use a semantic model to collect word co-occurrence information. In this paper we compare two such models: Hyperspace Analogue to Language (HAL) [42] and Bound Encoding of the Aggregate Language Environment (BEAGLE) [43]. These give us semantic smoothing matrices $\mathbf{H}$ and $\mathbf{B}$, respectively, to which we interchangeably refer to as $\mathbf{S}$. The matrix $\mathbf{H}$ is a square $N \times N$ matrix, while $\mathbf{B}$ is $N \times D$, where D is a chosen number of dimensions (defined below in the BEAGLE section).

The semantic information ($\mathbf{S}$) is multiplied with the sentence data and thus integrated into the kernel $\mathbf{K} = \kappa\ ((\mathbf{X} + \varepsilon)\mathbf{S}, (\mathbf{X} + \varepsilon)\mathbf{S})$. A small number $\varepsilon = 0.01$ is added to the training data to allow semantic smoothing across the whole feature set. The above approach has the effect of re-introducing the semantic information about the words, that was lost in the bag-of-words representation used to encode the features. Finally, by changing $\mathbf{S}$ and the kernel function κ, we are able to create different kernel matrices, which we then integrate using pMKL. In the following sections we describe pMKL and the semantic word co-occurrence models that comprise this methodology.

Probabilistic Multiple Kernel Learning

The data integration approach proposed and adopted in the present work belongs to the family of Multiple Kernel Learning (MKL) methods [44-47]. These approaches have recently gained significant attention due to their successful application in bioinformatics and pattern recognition domains [48,49] where multiple information sources are present.

In contrast with past ensemble approaches, such as classifier combination schemes where a separate model was trained on each individual source, MKL is a kernel-based data integration approach that attempts to informatively fuse the information sources *directly* within a single overall model. The intuition behind MKL and the difference from classifier combination methods is graphically depicted in Figure 2.

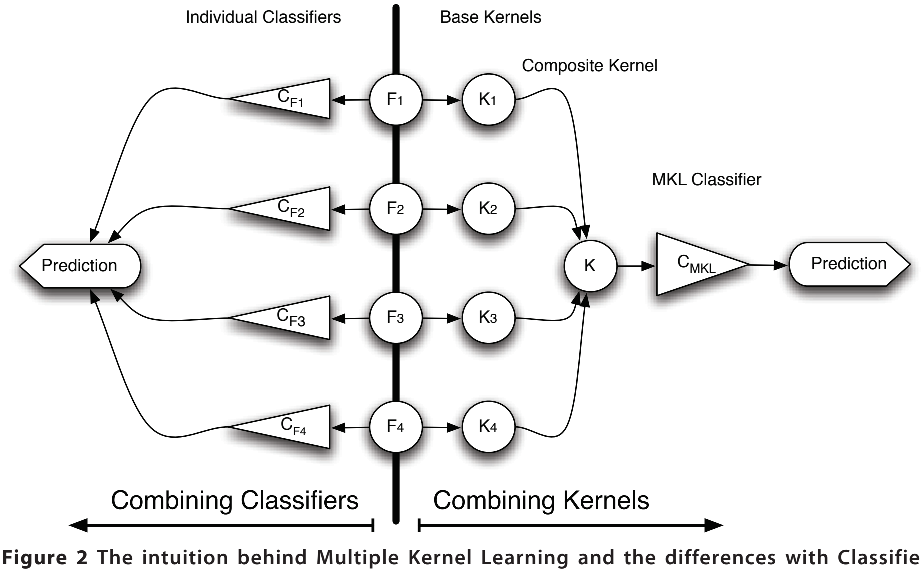


Figure 2 The intuition behind Multiple Kernel Learning and the differences with Classifier Combination methods. F_s denote the different feature spaces that are being combined, while the K_s are the different kernels which are contrasted with full classifiers C_{Fs}, shown on the left-hand side of the figure.

In this work, we employ the probabilistic MKL approach [50], proposed by [49], which follows a variational Bayesian formalism and results in probabilistic outputs capturing the model's uncertainty in class predictions. Individual semantic kernels, as analytically described in the next sections, are constructed from disparate unlabelled sources and then combined into an overall composite semantic kernel on which a single classifier operates.

The combination follows a parameterised convex linear rule, Equation 1, with kernel combination parameters β_s and individual kernel parameters $\boldsymbol{\theta}^{(s)}$ for the s semantic kernels (κ_s). Inference of the kernel combination parameters results in identification of an informative fusion of the base semantic kernels and, hence, also acts as a measure of their discriminative power. This will be crucial for selecting appropriate resolution levels for the base semantic kernels later on.

$$\begin{aligned}&\kappa\left(\mathbf{x}_i, \mathbf{x}_j, \boldsymbol{\beta}, \boldsymbol{\Theta}\right) = \sum_{s=1}^{S} \beta_s \kappa_s \left(\mathbf{x}_i^{(s)}, \mathbf{x}_j^{(s)}, \boldsymbol{\theta}^{(s)}\right) \\ &\text{with } \sum_{s=1}^{S} \beta_s = 1 \text{ and } \beta_s \geq 0 \ \forall \ s\end{aligned} \tag{1}$$

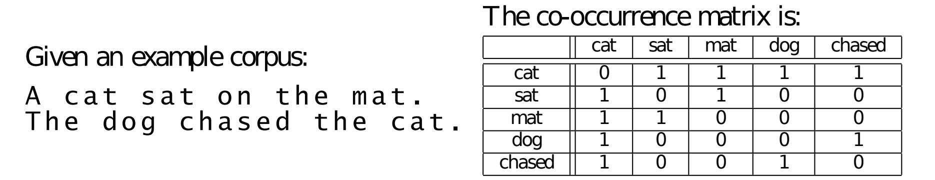
The overall MKL model is a Generalised Linear Model (GLM) [51] employing the multinomial probit likelihood within a variational Bayes approximation as described in [49]. In this work we concentrate on the construction and fusion of the base semantic kernels, which is the focal point of the next sections. The advantage of the adopted methodology is its probabilistic nature which allows a formal way to handle uncertainty.

Word co-occurrence models

Semantic models are representations of word meaning gathered statistically from large amounts of text. In general, they are constructed by considering each individual word in a corpus (referred to as a *target word*, or just *target*) and the text surrounding the word (called the *context*). The set of targets is denoted with T , while the set of context words, also referred to as *basis*, is B. The basis do not necessarily have to be words. The basis could also be grammatical structures, such as parse or dependency trees [52]. It is important that the basis match the feature type of the kernel. The product is a mapping of words into a multidimensional geometric space, in such a way that distance between the words corresponds to the distance of the word semantics according to their usage in text, Figure 3.

The main purpose of models is to calculate contextual similarity of words, consequently many of the models are constructed on the basis of two principles that reflect this goal. Firstly, in describing semantic models it is often said that "words are known by the company they keep", that is, the target's sense is defined by the surrounding words. Secondly, that this meaning can be learned automatically given enough examples of the usage of a word [53]. These two hypothesis have led to a great number of models, many of which have been validated both in psychological and linguistic experiments [42,43,52,54].

One of the main separating characteristics is the definition of the context used in the generation of the model. For example, Latent Semantic Analysis (LSA) [55] defines



Given an example corpus:

```
A cat sat on the mat.
The dog chased the cat.
```

The co-occurrence matrix is:

	cat	sat	mat	dog	chased
cat	0	1	1	1	1
sat	1	0	1	0	0
mat	1	1	0	0	0
dog	1	0	0	0	1
chased	1	0	0	1	0

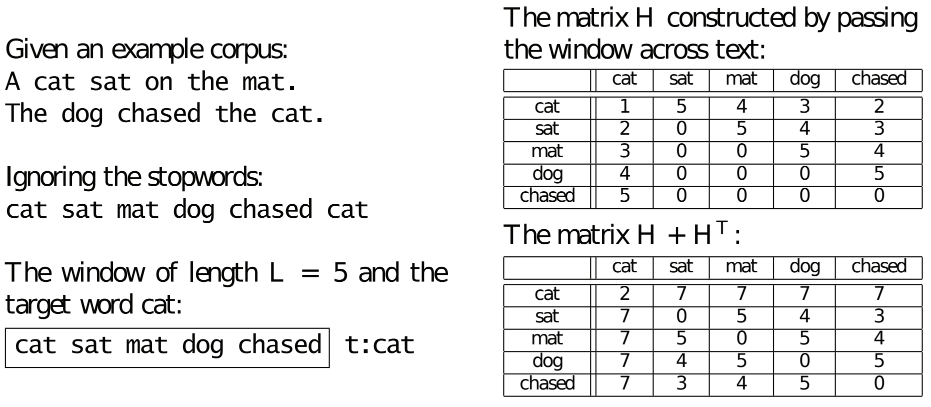
Figure 3 The vector space representation in a word-based semantic space. An example of the vector space representation in a word-based semantic space, where the context consists of all the words co-occurring with the target within the sentence. The columns represent the basis words that make up the contexts, while the rows are the target words. In this model the co-occurrence matrix is symmetric. The stop words (*a*, *the*, *on*) are ignored.

documents as contexts. Therefore, the words that are similar are the ones that can be found within same documents. On the other hand, [56] and [52], describe syntax-based models, where the context of the target is a path in the sentence dependency parse tree containing the word. Word co-occurrence models are the ones where the context consists of words immediately surrounding the target, within some specified window. Both HAL [42,57] and BEAGLE [43,58] are word-based models.

Word-based co-occurrence models are generally represented in the vector space. Each word corresponds to a vector whose dimensions are called the *basis*. In general, there exists a mapping between contexts and the basis. If this mapping is 1-to-1, the length of the target vectors is the number of all possible contexts. This can result in high-dimensional space corresponding to the number of unique words in the corpus. To limit the dimensionality and remove some noise, highly frequent function words are usually ignored. There are standard lists of these *stop words* containing most commonly occurring words including pronouns, determiners, and conjunctions. Depending on the final application of the model, the function words usually contain very little information.

HAL

Hyperspace Analogue to Language (HAL) is a semantic model that represents word similarity according to co-occurrence within a window of specific length [42,57,59]. The strength of word co-occurrence is determined by the distance between the two words within the specified window (Figure 4). This has the effect of boosting the



Given an example corpus:

```
A cat sat on the mat.
The dog chased the cat.
```

Ignoring the stopwords:

```
cat sat mat dog chased cat
```

The window of length L = 5 and the target word cat:

```
cat sat mat dog chased  t:cat
```

The matrix H constructed by passing the window across text:

	cat	sat	mat	dog	chased
cat	1	5	4	3	2
sat	2	0	5	4	3
mat	3	0	0	5	4
dog	4	0	0	0	5
chased	5	0	0	0	0

The matrix $H + H^T$:

	cat	sat	mat	dog	chased
cat	2	7	7	7	7
sat	7	0	5	4	3
mat	7	5	0	5	4
dog	7	4	5	0	5
chased	7	3	4	5	0

Figure 4 Construction of a HAL matrix. Construction of a HAL matrix from a small two-sentence corpus with the window of length L = 5. The stop words (*a*, *the*, *on*) are ignored.

similarity between words whose close contexts are the same, while allowing for variation in the phrasing of the context.

The $|T| \times |T|$ HAL matrix, $\mathbf{H}_o$, is constructed by passing a window of fixed length, L, across the corpus. The last word in the window is considered the target and the preceding words are the basis. Because the window slides across the corpus uniformly, the basis words are previous targets, and therefore the set of targets T is equivalent to the set of basis B, $T = B$.

The strength of the co-occurrence between a target and the basis depends on the distance between the two words, l, $1 \leq l \leq L$, within the window. The co-occurrence scoring formula, $L - l + 1$, assigns lower significance to words that are further apart. The overall co-occurrence of a target-basis pair is the sum of the scores assigned every time they coincide within the sliding window, across the whole corpus.

Even though the matrix is square, it is not symmetric. In fact, the transpose of the matrix reflects the co-occurrence scores with the basis that occur within the window of length L *after* the target. Thus $\mathbf{H}_o$ and $\mathbf{H}_o^T$ together reflect the full context (of length $2L - 1$) surrounding a target. There are two ways of combining this information so that it would be considered when the distance between targets is calculated. The first way is to concatenate $\mathbf{H}_{\text{o}}$ and $\mathbf{H}_o^T$ to produce a $|T| \times 2|B|$ matrix. The second way is to add the two matrices together $\mathbf{H}_o + \mathbf{H}_o^T$. Experimental testing showed that for our kernel combination method that the latter strategy is more effective. This was also the case when HAL was employed for query expansion [60]. Therefore, from now on when we refer to $\mathbf{H}$ we will assume $\mathbf{H} = \mathbf{H}_o + \mathbf{H}_o^T$.

BEAGLE

The Bound Encoding of the Aggregate Language Environment (BEAGLE) model [43,58] was proposed as a combined semantic space that incorporates word co-occurrence and word order. It is a word-based method where the context consists of words occurring in the same sentence as the target. Therefore, the set of targets and basis words is the same, and both consist of all unique words in the corpus. The data is stored in a vector space reduced by random mapping. If a context word appears frequently in the same sentence as a target word, its signal will be amplified through addition. Words sharing the same contexts will have strong signals corresponding to the common words.

Random mapping, sometimes also referred to as random projection or random indexing, is a method for reducing the dimensionality of data. For large data matrices, methods based on matrix decomposition such as principle component analysis (PCA) or singular value decomposition (SVD) can lead to heavy computational overheads [61-63]. On the other hand, random mapping provides a computationally efficient method of dimensionality reduction with minimal distortion in the distances between vectors [62]. It has been used for classification and clustering in a variety of applications including image and text [62,64], software quality [65], databases [66], and others [63].

The mapping transforms an $M \times N$ matrix, $\mathbf{X}$, into a lower dimensional space by multiplication with a $N \times D$ matrix of random values, $\mathbf{R}$. $\mathbf{R}$ can be constructed by random sampling from any distribution with the mean 0. The normalised rows form a near-orthogonal set of basis. The more dimensions are preserved, the more orthogonal the vectors are. In other words, the matrix $\mathbf{RR}^{\mathrm{T}} = \mathbf{I} + \varepsilon$, where ε is a small amount of noise that decreases as D increases [64].

Random mapping is used in BEAGLE in order to decouple the word vector lengths from the size of the vocabulary, as well as to reduce the vector length in order to allow for more efficient execution of costly matrix operations that are needed to encode word order [43].

The BEAGLE context matrix can be constructed by first building the $|T| \times |T|$ dimensional matrix of co-occurrence frequencies, such as in Figure 3, and subsequently reducing this space by multiplying it by a $|T| \times D$ matrix of random values, $\mathbf{R}$. Alternatively, it can be generated sequentially as the corpus is traversed. The latter method is more advantageous in that it allows for an expandable lexicon and it eliminates the need to store and transform the large frequency matrix. Addition of new words through corpus expansion only requires addition of new rows to the matrix.

The number of dimensions D is chosen so that it is large enough to ensure that this vector is unique for each target or basis word [58] suggest that multiples of 1024 are an appropriate choice for D, and use $D = 2048$ to encode larger corpora. Through empirical testing we found that $D = 4096$ gives us slightly better classification performance.

In this sequential method, each unique word in the corpus is assigned a D-dimensional vector of normally distributed random values drawn from the Gaussian distribution $\mathcal{N}(0, (\frac{1}{\sqrt{D}})^2)$. The choice of the standard deviation of $\frac{1}{\sqrt{D}}$ ensures normalised vector lengths. These are referred to as environmental vectors and denoted by $\mathbf{e}_b$, where b is a basis word. The $|T| \times D$ BEAGLE matrix, $\mathbf{B}$, where the rows are indexed by target words, is initialised to 0. The text is scanned in order, and for each target word t_i encountered, the context vector $\mathbf{c}_{t_i}$ for the current sentence s_k is calculated. $\mathbf{c}_{t_i}$ is the sum of the environmental vectors of the basis words, b_j, in the sentence. If we are only considering the contexts, the matrix entry for the target word t_i is the sum of the context vectors gathered form all the sentences s_k such that t_i occurs in s_k, $\mathbf{B}_{t_i} = \sum_{s_k} \mathbf{c}_{t_i}, \mathbf{c}_{t_i} = \sum \mathbf{e}_{b_j}$.

Experimental Setup

In our experiments we want to test the efficiency of combinations of semantic kernels by comparing them to single kernel results. In addition, we want to examine the potential of the weighted combinations of kernels to expand on our knowledge of the semantic methods.

For training data, we use the AImed data set [20], in which the protein entities are annotated and interacting pairs are specified, to judge which sentences contain interactions. The AImed corpus is emerging as standard and is being used in a variety of ways [14,21,23]. It consists of abstracts that contain PPI interactions, and have been annotated for proteins with a scheme that distinguishes the interacting protein pairs. It is, therefore, possible to separate the corpus into a data set that contains positive and negative example sentences. This can be done in two ways. For example, [20,21,23] separate the corpus into pairs of proteins, using the manually annotated protein entities. Interacting pairs are then used as positive training examples, while any two proteins, that occur in the same sentence and do not interact, constitute the negative data.

The other approach, and one which is employed in this paper, is to consider the sentences that contain interactions as positive examples, and the ones that do not, as negative. The reformulation of the problem has several advantages. This task is simpler to annotate than the full PPI, thus allowing for faster production of training data. Feature extraction does not require sentence parsing or preprocessing in a way that may be sensitive to annotation errors. The simpler classification leads to higher precision and recall, but only locates the sentence that describes the PPI and not the exact interacting pair. Thus while it is not fully automated, it might be more useful in a curation pipeline where the results need to be checked by humans [28]. Using the provided sentence segmentation, the data set contains 614 positive and 1355 negative sentences. All sentences are included regardless of the number of annotated proteins contained within.

When AImed data is used with the syntactic features, for example in [23,67], it is usually applied with the original 10-fold cross-validation (10 × 10 cv) data split provided by the dataset authors [20]. We can see from Figure 5, which demonstrates 10 different runs of a 10 cv experiment, that there can be great variations between the performance of an algorithm on different randomisations of data.

Therefore, it is more rigorous to run 10 different randomisations and all experiments are performed using ten times ten cross-validation (10 × 10 cv). In this way the training data is randomised, separated (as closely as possible) into ten equal parts. Nine of these parts are used for training and one for testing. This procedure is repeated with 10 different randomisations of the original data, providing 100 values for significance testing. In Figure 5, we show an illustration of two methods of data segmentation. In one method, the data is segmented so that no sentences in the test data come from the same abstract as a sentence in the training data. The abstract order is first randomised, the data is split into training and test portions as described above. The sentences are then further randomised within their set. In the other method, the

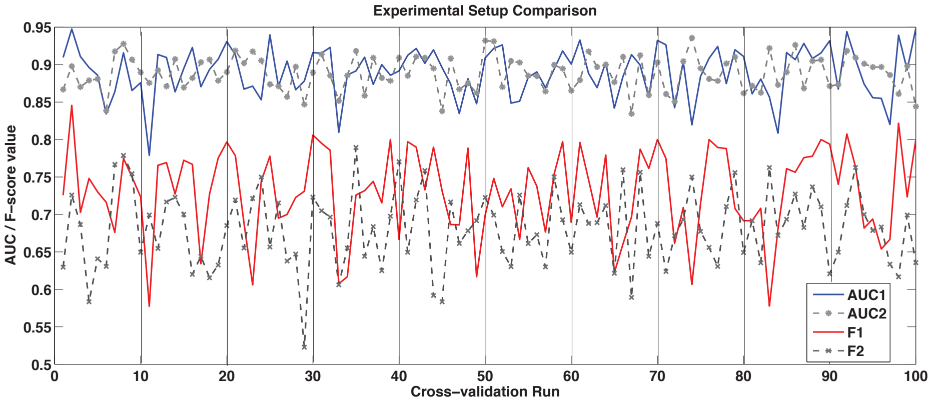


Figure 5 Comparison of experimental settings. This graph represents the AUC (higher pair of lines) and the F-score (lower pair of lines) results for two different experimental setups. In the first setup (AUC1, F1), all of the sentences from an abstract are either in training or test data within a fold of the cross-validation experiment. In the second setup, the sentence vectors are randomised first and then cut into cross-validation folds. The experiment shows 10 runs of randomised 10-fold cross-validation experiments. The vertical bars demonstrate the separations between the runs. For the first experiment F_1 = 0.7300 ± 0.0058 and AUC = 0.8902 ± 0.0033, while for the second experiment F_1 = 0.6808 ± 0.0051 and AUC = 0.8886 ± 0.0023. The t-test shows there is no significant difference between these experimental setups (p = 0.9640 for the F-scores and p = 0.5467 for the AUC).

sentences are randomised without observing the abstract of origin. In this particular experimental setup, the choice of randomisation technique provides no statistical difference across all of the experiments. The former method of sentence randomisation is also coupled with per-fold parameter tuning, while in the latter a single parameter is chosen for all of the folds. This may account for slightly better performance and the higher variance of the more stringent training-test data split. Significance testing is performed using a version of Student's t-test designed for 10 × 10 cross-validation [68].

We extract our training data from the AImed corpus in a manner similar to the vector space example in Figure 3. Each sentence is scanned and all the stop words [69] are removed. Each word is reduced to lowercase, any symbols or numbers are discarded, and the words are truncated to 10 letters [7]. Protein names are substituted by placeholder strings *PTNGNE* concatenated to the number of the protein within the sentence. This leaves 3,084 unique features. A sentence is then represented as a vector indexed by the unique features in the corpus. The number of times each feature occurs in the sentence is recorded in the vector. The anonymisation of proteins is likely to be one of the factors that minimises the effect of data randomisation methods (see Figure 5). The other factor is that sentence structure, and thus to some degree the authorship style, is disregarded.

The words in the corpora that were used as unlabelled data, GENIA and the subset of the Biomed Central open access articles [70] (OAA), are processed in the same way. GENIA is annotated for protein names, the OAA is not. So, for compatibility reasons, we have processed OAA with the Lingpipe sentence segmentation and named entity recognition software trained on GENIA. We used the *protein molecule* annotation as the indicator of protein presence. The OAA dataset also differs from AImed and GENIA in that it consists of full text articles, thus the results consist of a smaller portion of text and are described in a different, more detailed way. Similarities are created only for the words that occur in the training data. The HAL matrix is created as described in the previous section and in Figure 4, except that only the unique features from AImed are considered as targets and basis. This leads to a sparse $N \times N$ matrix $\mathbf{H}$, which is then multiplied with $\mathbf{X}$. On the other hand, for the BEAGLE matrix we still only consider AImed training features as targets; however, the bases consist of all words that co-occur in the sentences with these targets. For GENIA there are around 12,000 basis, and for OAA only the first 30,000 basis are considered, but the random projection technique keeps the BEAGLE matrix size consistent at $N \times D$ dimensions, where $D = 4096$.

We measure the efficiency of classification using the AUC and F-score measures. The AUC is the area under the receiver operating characteristic (ROC), which depicts the true positive rate vs. the false positive rate of a classifier's testing output. The closer the AUC is to 1, the better the classification results. The F-score is often used in evaluating natural language processing tasks. It is a balanced measure of precision (P) and recall (R). Here we use $F_1 = \frac{2PR}{P+R}$. The error is defined as the percentage of testing points that were wrongly classified.

Finally, the pMKL algorithm has no parameters akin to the SVM regularisation parameter. The only parameters that required tuning were the Gaussian kernel parameter and the HAL window size parameter. The Gaussian kernel parameter that produces the highest AUC with both HAL and BEAGLE kernels is determined using 1 × 3 cross-validation at each fold of the 10 × 10 cv experiment (Figure 6). The range of

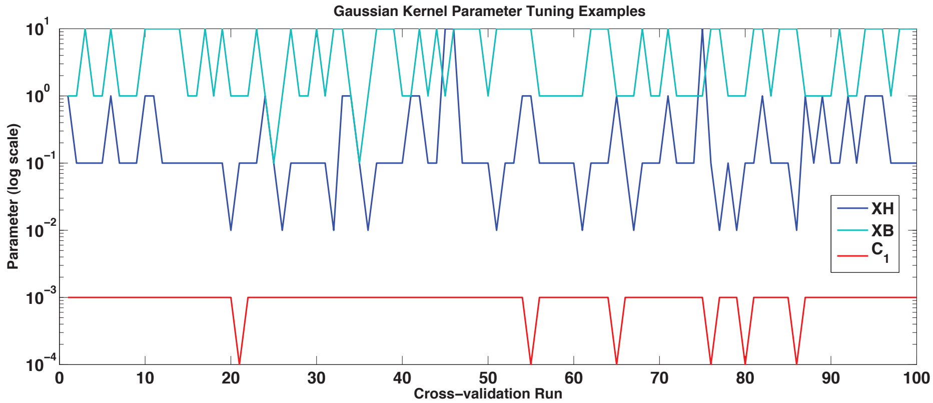


Figure 6 Kernel Parameter Estimation. This graph shows the parameters that were chosen for some of the experiments in this paper. The top two lines describe experiments with the OAA dataset whose results are shown in Table 1. While the third line describes the parameters chosen for the combined kernel $\mathbf{C}_1$ from Table 2.

values examined are the powers of 10 between 10^{-5} and 10. For the combinations of composite HAL kernels the preferred values tended towards the smaller parameters, for plain data the parameter chosen was 0.1 for 99 of the 100 folds, while for data transformed by a single HAL or BEAGLE kernel, the values ranged in the set (0.1, 1, 10). The approach of tuning parameters at each fold is time consuming, but 1 × 3 cv performs just as well as 2 × 5 cv, in this experiment. There is also no statistical difference between choosing the best result out of several parallel 10 × 10 cv experiments, each run with a particular assigned kernel parameter, and the above per-fold tuning method. The polynomial kernel parameter is 2.

Results and Discussion

The purpose of the experiments in this paper is to verify that using combinations of multiple semantic kernels can improve classification performance. We do this with the simplest possible features, in order to avoid introducing further complexity. As a result, it is difficult to compare the results to full PPI extraction tasks, so we provide single kernel baseline results. In the general, the pMKL algorithm produces results comparable to the SVM, but without the need to tune the extra margin parameter. Depending on the task, features, kernel, kernel parameter, and margin parameter choice the pMKL, GPs, and the SVM might slightly, but significantly outperform each other, but in general will provide similar results [40]. This section is divided in three parts. In the first part, we provide the baseline results using plain and semantic kernels. Secondly, we examine many fixed combinations of the basic kernels, and report the best combinations. In the final part, we examine the effectiveness of the pMKL's ability to estimate the best weighted sum of the kernels, by observing the changes in the predictive likelihood. This estimation is done without observation of the true labels of the test data, and therefore might not lead to the optimal F-score or AUC.

Single kernel results

The baseline for the evaluation of pMKL is provided through single kernel experiments, the results of which are provided in Table 1. These results are consistent with the semantic kernel experiments performed with the GP classifiers in [34]. The

Table 1 Results of the pMKL single kernel experiments

Kernel	F-score	Error	Precision	Recall	AUC
$\mathbf{C}_0$: **X**[a]	0.7300 ± 0.0058	17.6878 ± 0.3158	0.6893 ± 0.0072	0.7828 ± 0.0079	0.8902 ± 0.0033
XH[b]	0.7060 ± 0.0060	18.3453 ± 0.3203	0.6989 ± 0.0074	0.7210 ± 0.0084	0.8899 ± 0.0031
XB[b]	0.6567 ± 0.0080	20.6249 ± 0.4113	0.6759 ± 0.0086	0.6501 ± 0.0113	0.8776 ± 0.0035
XH[c]	0.6921 ± 0.0056	18.5716 ± 0.3200	0.7113 ± 0.0072	0.6808 ± 0.0077	0.8888 ± 0.0029
XB[c]	0.7267 ± 0.0052	17.2958 ± 0.3087	0.7117 ± 0.0070	0.7490 ± 0.0071	**0.9000* ± 0.0028**

[a] The original data with the Gaussian kernel.

[b] The data smoothed with the HAL and BEAGLE (Gaussian) matrices created from GENIA dataset.

[c] The data smoothed with the HAL and BEAGLE (Gaussian) matrices created from the OAA dataset.

* Statistically significant (*p* = 0.0038) compared to kernel C_0.

ultimate baseline is provided by using pMKL with a plain Gaussian kernel, which produces higher F-score and AUC than the cosine and polynomial kernels.

We then add the Gaussian semantic kernels created using HAL and BEAGLE. We find that with pMKL, unlike with GPs, the smoothing using the HAL matrix produces a slight reduction in AUC over the plain Gaussian kernel, although this is not statistically significant. While there is little difference between the results produced with the HAL kernel created from Genia or OAA, the larger data set produced significantly better results when applied with the BEAGLE method.

In these experiments $\mathbf{H}$ was constructed from $l = 1$ which was shown to lead to the best results with this data and GPs [34]. Under the speculation that more data is better than hand annotated data, we proceed with multiple kernel experiments using the OAA dataset.

Multiple kernel results

We perform two types of multiple kernel experiments. In the first kind we evaluate uniform compositions of multiple kernels, we then estimate combinations of different kernels in order to gain insight into their predictive properties. Many different combinations of semantic kernels could be formed, so the following experiments are illustration of possible uses.

Table 2 shows that combinations of kernels can lead to a statistically significant increase in the AUC. As the parameter tuning was performed with the observation of the maximum AUC, the results reflect that. For experiments where the F-score is the primary concern, the tuning should be performed by observing the highest F-score. While Figure 5 demonstrates the general trend of the two measures is similar, the tuning strategy can make a difference in the outcome. The kernel combination $\mathbf{C}_1$ is the uniform weighting of the HAL matrices at different window lengths together this

Table 2 Results of the pMKL multiple kernel experiments with fixed weights

Kernel	F-score	Error	Precision	Recall	AUC
$\mathbf{C}_1$	0.7039 ± 0.0054	17.5614 ± 0.3378	0.7381 ± 0.0059	0.6790 ± 0.0080	0.8881 ± 0.0029
$\mathbf{C}_2$	0.7052 ± 0.0054	18.3698 ± 0.3337	0.7012 ± 0.0063	0.7155 ± 0.0076	0.8838 ± 0.0031
$\mathbf{C}_3$	0.7359 ± 0.0045	16.8581 ± 0.2934	0.7156 ± 0.0063	0.7631 ± 0.0063	**0.9092* ± 0.0023**
$\mathbf{C}_4$	0.6633 ± 0.0080	19.0861 ± 0.3507	0.7301 ± 0.0078	0.6242 ± 0.0119	0.8883 ± 0.0029

C_1: Uniform sum of Gaussian kernels from HAL matrices with window lengths *l* 1 through 10 (OAA).

C_2: Uniform sum of Gaussian kernels from HAL matrices with window lengths *L* 1 through 10 (OAA).

C_3: Uniform sum of 6 Gaussian, cosine, and polynomial kernels using HAL $H_{l=1}$ from both OAA and GENIA.

C_4: Estimated sum of Gaussian kernels from HAL matrices with window lengths *l* 1 through 10 (OAA).

*Statistically significant (*p* = 0.0016) compared to kernel C_0.

weighting is equivalent to $\sum_{l=1}^{10} \mathbf{H}_l$. We find that there is no difference in performance over smoothing with the single kernel $\mathbf{H}_{l\ =\ 1}$ from Table 1. The kernel combination $\mathbf{C}_2$ is the uniform sum of combined HAL matrices $\sum_{L=1}^{10} \mathbf{H}_L$, where $\mathbf{H}_L = \sum_{l=1}^{L} (L - l + 1)\mathbf{H}_l$. This combination contains redundant information over $\mathbf{C}_1$, but this strategy provides only a decrease in performance. However, $\mathbf{C}_3$, which is a combination of different views of the data using different kernel types and combinations of both GENIA and OAA $\mathbf{H}_{l\ =\ 1}$ smoothing provides an increase in performance. This indicates that the performance gain is best achieved when combining kernels that contain diverse information or at least diverse views of that information.

Estimating the kernel weights

Although the best results come from fixed kernel weights, we can gain significant insight into the predictive quality of the data by exploiting the pMKL kernel weight estimation property.

In particular, we are interested in examining the properties of HAL matrices. These matrices are composites and each matrix $\mathbf{H}$ created with context length L can be considered a combination of L matrices, such that $\mathbf{H} = \sum_{l=1}^{L} (L - l + 1)\mathbf{H}_l$. Therefore, in addition to the right choice of kernels and kernel settings we need to make the right choice of L. There is also a dispute over the weighting function (L - l + 1); for example, [71] found that using a uniform weighting as opposed to a decaying one produces better search query expansion results.

Figure 7 shows the estimated weights for kernels constructed with $\mathbf{XH}_l$ for l = 1 ... 10 ($\mathbf{C}_4$, in Table 2). The assigned weightings closely mirror the sparsity of the HAL matrices. Matrices 2 and 3 have the lowest sparsity, and while the contribution of l = 1 seems to be underestimated, l = 3 seems to be overestimated. This would indicate that, perhaps, a scheme weighted by the information stored in matrices representing various window lengths would lead to best performance when applying the HAL algorithm to various tasks. The AUC (0.8883 ± 0.0029) is slightly higher than the uniform combination of these kernels while the F-score (0.6633 ± 0.0080) is significantly lower than in uniform combination of these kernels ($\mathbf{C}_1$, in Table 2). Due to the computationally intensive nature of this experiment the parameters for each

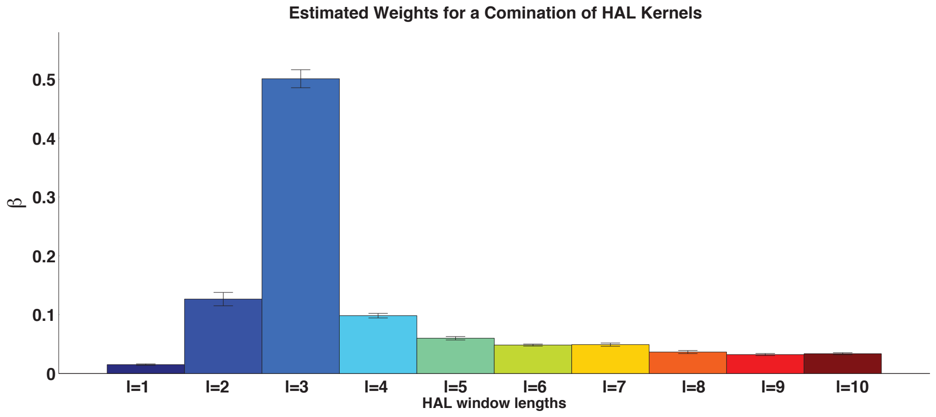


Figure 7 The weights assigned by pMKL to words at different distances from the target word. The bars represent the kernels constructed from HAL matrices with, from left to right, *l* = 1 to *l* = 10.

of the kernels was set at 1, which is one of the parameters the cross-validated tuning approach favoured for $\mathbf{XH}_{l=1}$ (Figure 6).

Conclusions

This paper describes a smoothing approach, which is similar to the methods using semantic kernels created from WordNet [72] or Wikipedia information [73,74]. However, this method provides a domain-independent alternative, by using automatically derived semantic information for classification. It also gives an application-based way of evaluating the quality of word co-occurrence matrices, which is a difficult task usually requiring specialised human judgements.

The results presented in this paper show that using combinations of kernels can lead to significant improvement in both F-score and AUC. In addition, we are able to use pMKL kernel weight estimation for kernel selection as well as for gaining important insights into the quality and linguistic properties of the data. This is an introduction to this approach, which uses simple features that do not require dependency parsing, and thus is not directly comparable to the full extraction methods that are popularly used with PPI data sets. In the future we will investigate this method with semantic models that are compatible with dependency-based features, and kernels.

Acknowledgements
TP was funded by a Scottish Enterprise PhD studentship. TD acknowledges funding from the NSF Expeditions in Computing grant on Computational Sustainability (Award Number 0832782). MG was funded by an EPSRC Advanced Research Fellowship EP/E052029/1 and EPSRC project CLIMB EP/F009429/1.

Author details
[1]School of Computing Science, University of Glasgow, Glasgow, UK. [2]Department of Computer Science, Cornell University, 14850, Ithaca, NY, US. [3]Department of Statistical Science, University College London, London, UK.

Authors' contributions
TP and MG designed the semantic kernel method. TD and MG designed the pMKL approach, which was implemented by TD. TP implemented the semantic kernel method and ran the experiments in this paper. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Received: 26 October 2010 Accepted: 14 May 2011 Published: 14 May 2011

References

1. Young KH: **Yeast two-hybrid: so many interactions, (in) so little time.** *Biol Reprod* 1998, **58(2)**:302-311[http://www.hubmed.org/display.cgi?uids=9475380].
2. Nanas N, Roeck AND, Vavalis M: **What Happened to Content-Based Information Filtering?** In *ICTIR 2009, Volume 5766 of Lecture Notes in Computer Science.* Edited by: Azzopardi L, Kazai G, Robertson SE, Rüger SM, Shokouhi M, Song D, Yilmaz E. Springer; 2009:249-256.
3. Kanehisa M, Goto S, Furumichi M, Tanabe M, Hirakawa M: **KEGG for representation and analysis of molecular networks involving diseases and drugs.** *Nucleic Acids Res* 2010, , **38 Database:** 355-360[http://www.hubmed.org/display.cgi?uids=19880382].
4. Mewes HW, Amid C, Arnold R, Frishman D, Güldener U, Mannhaupt G, Münsterkötter M, Pagel P, Strack N, Stümpen V, Warfsmann J, Ruepp A: **MIPS: analysis and annotation of proteins from whole genomes.** *Nucleic Acids Res* 2004, , **32 Database:** 41-44[http://www.hubmed.org/display.cgi?uids=14681354].
5. Alfarano C, Andrade CE, Anthony K, Bahroos N, Bajec M, Bantoft K, Betel D, Bobechko B, Boutilier K, Burgess E, Buzadzija K, Cavero R, D'Abreo C, Donaldson I, Dorairajoo D, Dumontier MJ, Dumontier MR, Earles V, Farrall R, Feldman H, Garderman E, Gong Y, Gonzaga R, Grytsan V, Gryz E, Gu V, Haldorsen E, Halupa A, Haw R, Hrvojic A, Hurrell L, Isserlin R, Jack F, Juma F, Khan A, Kon T, Konopinsky S, Le V, Lee E, Ling S, Magidin M, Moniakis J, Montojo J, Moore S, Muskat B, Ng I, Paraiso JP, Parker B, Pintilie G, Pirone R, Salama JJ, Sgro S, Shan T, Shu Y, Siew J, Skinner D, Snyder K, Stasiuk R, Strumpf D, Tuekam B, Tao S, Wang Z, White M, Willis R, Wolting C, Wong S, Wrong A, Xin C, Yao R, Yates B, Zhang S, Zheng K, Pawson T, Ouellette BF, Hogue CW: **The Biomolecular Interaction Network Database and related tools 2005 update.** *Nucleic Acids Res* 2005, , **33 Database:** 418-424[http://www.hubmed.org/display.cgi?uids=15608229].
6. Cohen AM, Hersh WR: **A survey of current work in biomedical text mining.** *Briefings in Bioinformatics* 2005, **6**:51-71.

7. Donaldson I, Martin J, de Bruijn B, Wolting C, Lay V, Tuekam B, Zhang S, Baskin B, Bader GD, Michalickova K, Pawson T, Hogue CW: **PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine.** *BMC Bioinformatics* 2003, **4(11)**[http://www.biomedcentral.com/1471-2105/4/11].
8. Alex B, Grover C, Haddow B, Kabadjov M, Klein E, Matthews M, Roebuck S, Tobin R, Wang X: **Assisted Curation: Does Text Mining Really Help?** *Proceedings of Pacific Symposium on Biocomputing, Hawaii, USA* 2008.
9. Jenssen TK, Laegreid A, Komorowski J, Hovig E: **A literature network of human genes for high-throughput analysis of gene expression.** *Nat Genet* 2001, **28**:21-28[http://www.hubmed.org/display.cgi?uids=11326270].
10. Chen H, Sharp BM: **Content-rich biological network constructed by mining PubMed abstracts.** *BMC Bioinformatics* 2004, **5**:147-147[http://www.hubmed.org/display.cgi?uids=15473905].
11. Rinaldi F, Schneider G, Kaljurand K, Hess M, Andronis C, Konstandi O, Persidis A: **Mining of relations between proteins over biomedical scientific literature using a deep-linguistic approach.** *Artif Intell Med* 2007, **39(2)**:127-136[http://www.hubmed.org/display.cgi?uids=17052900].
12. Koster C, Seibert O, Seutter M: **The PHASAR Search Engine.** *Proceedings NLDB 2006, Springer LNCS 3999* 2006, 141-152.
13. Marcotte EM, Xenarios I, Eisenberg D: **Mining literature for protein-protein interactions.** *Bioinformatics* 2001, **17**:359-363.
14. Giuliano C, Lavelli A, Romano L: **Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature.** *Proceedings of EACL 2006* 2006, 401-408.
15. Rosario B, Hearst M: **Multi-way Relation Classification: Application to Protein-Protein Interaction.** *Proceedings of HLT-NAACL'05* 2005 [http://biotext.berkeley.edu/papers/hlt-emnlp05-rosario.pdf].
16. Thomas J, Milward D, Ouzounis C, Pulman S, Carroll M: **Automatic Extraction of Protein Interactions from Scientific Abstracts.** *Pacific Symposium on Biocomputing 5* 2000, 538-549.
17. Miyao Y, Ohta T, Masuda K, Tsuruoka Y, Yoshida K, Ninomiya T, Tsujii J: **Semantic retrieval for the accurate identification of relational concepts in massive textbases.** *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics* Morristown, NJ, USA: Association for Computational Linguistics; 2006, 1017-1024.
18. Rebholz-Schuhmann D, Kirsch H, Arregui M, Gaudan S, Riethoven M, Stoehr P: **EBIMed-text crunching to gather facts for proteins from Medline.** *Bioinformatics* 2007, **23(2)**:237-244[http://www.hubmed.org/display.cgi?uids=17237098].
19. Katrenko S, Adriaans P: **Learning Relations from Biomedical Corpora Using Dependency Trees.** *Knowledge Discovery and Emergent Complexity in BioInformatics, Lecture Notes in Computer Science* 2006, **4366.**
20. Bunescu R, Ge R, Kate RJ, Marcotte EM, Mooney RJ, Ramani AK, Wong YW: **Comparative experiments on learning information extractors for proteins and their interactions.** *Artif Intell Med* 2005, **33(2)**:139-155[http://www.hubmed.org/display.cgi?uids=15811782].
21. Erkan G, Özgür A, Radev D: **Semi-Supervised Classifition for Extracting Protein Interaction Sentences using Dependency Parsing.** *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)* 2007, 228-237[http://www.aclweb.org/anthology/D/D07/D07-1024].
22. Cussens J, Nédellec C, (Eds): *Proceedings of the 4th Learning Language in Logic Workshop (LLL05), Bonn* 2005 [http://www.cs.york.ac.uk/aig/lll/lll05/proceedings.pdf].
23. Airola A, Pyysalo S, Björne J, Pahikkala T, Ginter F, Salakoski T: **All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning.** *BMC bioinformatics* 2008, **9(Suppl 11)**[http://dx.doi.org/10.1186/1471-2105-9-S11-S2].
24. Pyysalo S, Airola A, Heimonen J, Björne J, Ginter F, Salakoski T: **Comparative analysis of five protein-protein interaction corpora.** *BMC Bioinformatics* 2008, **9(Suppl 3)**:S6[http://www.biomedcentral.com/1471-2105/9/S3/S6].
25. Lewis DD: **Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval.** *ECML '98: Proceedings of the 10th European Conference on Machine Learning* London, UK: Springer-Verlag; 1998, 4-15.
26. Tikk D, Thomas P, Palaga P, Hakenberg J, Leser U: **A Comprehensive Benchmark of Kernel Methods to Extract Protein-Protein Interactions from Literature.** *PLoS Comput Biol* 2010, **6(7)**:e1000837[http://dx.doi.org/10.1371%2Fjournal.pcbi.1000837].
27. Clegg AB, Shepherd AJ: **Benchmarking natural-language parsers for biological applications using dependency graphs.** *BMC Bioinformatics* 2007, **8**:24-24[http://www.hubmed.org/display.cgi?uids=17254351].
28. Albert S, Gaudan S, Knigge H, Raetsch A, Delgado A, Huhse B, Kirsch H, Albers M, Rebholz-Schuhmann D, Koegl M: **Computer-assisted generation of a protein-interaction database for nuclear receptors.** *Journal of Molecular Endocrinology* 2003, **8(17)**:1555-67[http://www.ebi.ac.uk/Rebholz/publications.html].
29. Hirschman L, Morgan AA, Yeh AS: **Rutabaga by any other name: extracting biological names.** *J Biomed Inform* 2002, **35(4)**:247-259[http://www.hubmed.org/display.cgi?uids=12755519].
30. Subramaniam LV, Mukherjea S, Kankar P, Srivastava B, Batra VS, Kamesam PV, Kothari R: **Information extraction from biomedical literature: methodology, evaluation and an application.** *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management* New York, NY, USA: ACM Press; 2003, 410-417.
31. Tanabe L, Xie N, Thom LH, Matten W, Wilbur WJ: **GENETAG: a tagged corpus for gene/protein named entity recognition.** *BMC Bioinformatics* 2005, **6(Suppl 1)**[http://www.hubmed.org/display.cgi?uids=15960837].
32. Alex B, Haddow B, Grover C: **Recognising Nested Named Entities in Biomedical Text.** *Biological, translational, and clinical language processing, Prague, Czech Republic: Association for Computational Linguistics* 2007, 65-72[http://www.aclweb.org/anthology/W/W07/W07-1009].
33. Smith L, Tanabe LK, Ando RJ, Kuo CJ, Chung IF, Hsu CN, Lin YS, Klinger R, Friedrich CM, Ganchev K, Torii M, Liu H, Haddow B, Struble CA, Povinelli RJ, Vlachos A, Baumgartner WA, Hunter L, Carpenter B, Tsai RT, Dai HJ, Liu F, Chen Y, Sun C, Katrenko S, Adriaans P, Blaschke C, Torres R, Neves M, Nakov P, Divoli A, Maña-López M, Mata J, Wilbur WJ: **Overview of BioCreative II gene mention recognition.** *Genome Biol* 2008, **9(Suppl 2)**[http://www.hubmed.org/display.cgi?uids=18834493].
34. Polajnar T, Rogers S, Girolami M: **Classification of Protein Interaction Sentences via Gaussian Processes.** *Proceedings of 4th IAPR International Conference, Pattern Recognition in Bioinformatics* Springer Verlag; 2009, 282-292.
35. Sekimizu T, Park H, Tsujii J: **Identifying the interaction between genes and gene products based on frequently seen verbs MEDLINE abstracts.** 1998 [http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.66].

36. Friedman C, Kra P, Yu H, Krauthammer M, Rzhetsky A: **GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles.** *Bioinformatics* 2001, **17(Suppl 1)**:S74-S82.
37. Polajnar T, Girolami M: **Semi-supervised Prediction of Protein Interaction Sentences Exploiting Semantically Encoded Metrics.** *Proceedings of the 4th IAPR International Conference, Pattern Recognition in Bioinformatics* Springer Verlag; 2009, 270-281.
38. Chapelle O, Schölkopf B, Zien A, (Eds): *Semi-Supervised Learning* Cambridge, MA: MIT Press; 2006 [http://www.kyb.tuebingen.mpg.de/ssl-book].
39. Abney S: *Semisupervised Learning for Computational Linguistics* Chapman & Hall/CRC; 2007.
40. Polajnar T: **Semantic Models as Metrics for Kernel-based Interaction Identification.** *PhD thesis* University of Glasgow; 2010.
41. Kim JD, Ohta T, Tateisi Y, Tsujii J: **GENIA corpus-semantically annotated corpus for bio-textmining.** *Bioinformatics* 2003, **19(Suppl 1)**:180-182[http://www.hubmed.org/display.cgi?uids=12855455].
42. Lund K, Burgess C: **Producing high-dimensional semantic spaces from lexical CO-occurrence.** *Behavior Research Methods, Instrumentation, and Computers* 1996, **28(2)**:203-208.
43. Jones MN, Mewhort DJK: **Representing word meaning and order information in a composite holographic lexicon.** *Psychological Review* 2007, **114**:1-37.
44. Lanckriet GRG, Cristianini N, Bartlett P, Ghaoui LE, Jordan MI: **Learning the Kernel Matrix with Semidefinite Programming.** *Journal of Machine Learning Research* 2004, **5**:27-72.
45. Bach F, Lanckriet G, Jordan M: **Multiple kernel learning, conic duality, and the SMO algorithm.** *ICML '04: Proceedings of the twenty-first international conference on Machine learning, ACM* 2004.
46. Girolami M, Rogers S: **Hierarchic Bayesian Models for Kernel Learning.** *ICML '05: Proceedings of the 22nd international conference on Machine learning* New York, NY, USA: ACM; 2005, 241-248.
47. Bach F: **Exploring large feature spaces with hierarchical multiple kernel learning.** *Advances in neural information processing systems 20: proceedings of the 2007 conference* 2008.
48. Lanckriet GRG, Bie TD, Cristianini N, Jordan MI, Noble WS: **A statistical framework for genomic data fusion.** *Bioinformatics* 2004, **20(16)**:2626-2635.
49. Damoulas T, Girolami MA: **Probabilistic multi-class multi-kernel learning: On protein fold recognition and remote homology detection.** *Bioinformatics* 2008 [http://www.hubmed.org/display.cgi?uids=18378524].
50. **PMKL Software.** [http://www.dcs.gla.ac.uk/inference/pMKL].
51. McCullagh P, Nelder JA: *Generalised Linear Models* London: Chapman & Hall; 1989.
52. Padó S, Lapata M: **Dependency-Based Construction of Semantic Space Models.** *Comput Linguist* 2007, **33(2)**:161-199.
53. McMurray B: **Moo-cow! Mummy! More! How do children learn so many words?** *Significance* 2007, **4(4)**:159-163.
54. Lowe W: **Towards a theory of semantic space.** In *Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society.* Edited by: Moore JD, Stenning K. Mahwah NJ: Lawrence Erlbaum Associates; 2001:576-581.
55. Landauer TK, Foltz PW, Laham D: **An Introduction to Latent Semantic Analysis.** *Discourse Processes* 1998, **25**:259-284.
56. Lin D: **Automatic Retrieval and Clustering of Similar Words.** *Proceedings of the Joint Annual Meeting of the Association for Computational Linguistics and International Conference on Computational Linguistics, Montreal, Canada* 1998, 768-774.
57. Burgess C, Livesay K, Lund K: **Explorations in context space: Words, sentences, discourse.** *Discourse Processes* 1998, **25**:211-257.
58. Jones MN, Kintsch W, Mewhort DJ: **High-dimensional semantic space accounts of priming.** *Journal of Memory and Language* 2006, **55(4)**:534-552[http://dx.doi.org/10.1016/j.jml.2006.07.003].
59. Burgess C, Lund K: **Modeling parsing constraints with high-dimensional context space.** *Language and Cognitive Processes* 1997, **12**:177-210.
60. Song D, Bruza PD: **Discovering Information Flow Using a High Dimensional Conceptual Space.** *Proceedings of ACM SIGIR 2001* 2001, 327-333.
61. Papadimitriou CH, Raghavan P, Tamaki H, Vempala S: **Latent semantic indexing: a probabilistic analysis.** *Journal of Computer and System Sciences* 2000, **61(2)**:217-235.
62. Bingham E, Mannila H: **Random projection in dimensionality reduction: applications to image and text data.** *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* New York, NY, USA: ACM; 2001, 245-250.
63. Fradkin D, Madigan D: **Experiments with random projections for machine learning.** *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* New York, NY, USA: ACM; 2003, 517-522.
64. Kaski S: **Dimensionality reduction by random mapping: fast similarity computation for clustering.** *Proceedings of the 1998 IEEE International Joint Conference on Neural Networks* 1998, **1**:413-418.
65. Jin X, Bie R: **Improving Software Quality Classification with Random Projection.** *Proceedings of ICCI 2006: 5th IEEE International Conference on Cognitive Informatics* 2006, **1**:149-154.
66. Achlioptas D: **Database-friendly random projections.** *PODS '01: Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* New York, NY, USA: ACM; 2001, 274-281.
67. Saetre R, Sagae K, Tsujii J: **Syntactic features for protein-protein interaction extraction.** *Proceedings of LBM'07, volume 319* 2008.
68. Bouckaert RR, Frank E: **Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms.** In *Advances in Knowledge Discovery and Data Mining, Volume 3056 of Lecture Notes in Computer Science.* Edited by: Dai H, Srikant R, Zhang C. Springer Berlin/Heidelberg; 2004:3-12[http://dx.doi.org/10.1007/978-3-540-24775-3%5F3].
69. **Cornell University Stop Word List.** [ftp://ftp.cs.cornell.edu/pub/smart/english.stop].
70. **Biomed Central Open Access Articles.** [http://www.biomedcentral.com/info/about/datamining/].
71. Azzopardi L, Girolami M, Crowe M: **Probabilistic hyperspace analogue to language.** *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* New York, NY, USA: ACM; 2005, 575-576.
72. Fellbaum C, (Ed): *WordNet: An Electronic Lexical Database* Cambridge, MA: MIT Press; 1998.
73. Basili R, Cammisa M, Moschitti A: **A Semantic Kernel to Exploit Linguistic Knowledge.** *AI*IA 2005: Advances in Artificial Intelligence* 2005, 290-302.

74. Minier Z, Bodo Z, Csato L: **Wikipedia-Based Kernels for Text Categorization.** *SYNASC '07: Proceedings of the Ninth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing* Washington, DC, USA: IEEE Computer Society; 2007, 157-164.

doi:10.1186/2041-1480-2-1
Cite this article as: Polajnar *et al.*: Protein interaction sentence detection using multiple semantic kernels. *Journal of Biomedical Semantics* 2011 **2**:1.