A Statistical Relational Learning Approach to Identifying Evidence Based Medicine Categories

Mathias Verbeke[◇] **Vincent Van Asch**[♣] **Roser Morante**[♣]

Paolo Frasconi[♠] **Walter Daelemans**[♣] **Luc De Raedt**[◇]

[◇] Department of Computer Science, Katholieke Universiteit Leuven, Belgium
{mathias.verbeke, luc.deraedt}@cs.kuleuven.be
[♣] Department of Linguistics, Universiteit Antwerpen, Belgium
{roser.morante, vincent.vanasch, walter.daelemans}@ua.ac.be
[♠] Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze, Italy
p-f@dsi.unifi.it

Abstract

Evidence-based medicine is an approach whereby clinical decisions are supported by the best available findings gained from scientific research. This requires efficient access to such evidence. To this end, abstracts in evidence-based medicine can be labeled using a set of predefined medical categories, the so-called *PICO* criteria. This paper presents an approach to automatically annotate sentences in medical abstracts with these labels. Since both structural and sequential information are important for this classification task, we use *kLog*, a new language for statistical relational learning with kernels. Our results show a clear improvement with respect to state-of-the-art systems.

1 Introduction

Evidence-based medicine (EBM) or *evidence-based practice (EBP)* combines clinical expertise, the preferences and values of the patient and the best available evidence to make good patient care decisions. Clinical research findings are systematically reviewed, appraised and used to improve the patient care, for which efficient access to such evidence is required. In order to facilitate the search process, medical documents are labeled using a set of predefined medical categories, the *PICO criteria*. PICO is an acronym for the mnemonic concepts that are used to construct queries when searching for scientific evidence in the EBM process. The need to automatize the annotation process has initiated research into automatic approaches to annotate sentences in medical documents with the PICO labels.

As indicated by Kim et al. (2011), both the structural information of the words in the sentence, and that of the sentences in the document are important features for this task. Furthermore, sequential information can leverage the dependencies between different sentences in the text. Therefore we propose an approach using *kLog* (Frasconi et al., 2012) to tackle this problem. kLog is a new language for statistical relational learning with kernels, that is embedded in Prolog, and builds upon and links together concepts from database theory, logic programming and learning from interpretations. Learning from interpretations is a logical and relational learning setting (De Raedt et al., 2008) in which the examples are interpretations, that is, sets of tuples that are true in the examples. In a sense, each example can be viewed as a small relational database. kLog is able to transform relational into graph-based representations and apply kernel methods to extract an extended high-dimensional feature space.

The choice for kLog was motivated by previous results (Verbeke et al., 2012), where we showed that a statistical relational learning approach using kLog is able to process the contextual aspects of language improving on state-of-the-art results for hedge cue detection. However, the current task adds two levels of complexity. First, next to the relations between the words in the sentence, now also the relations between the sentences in the document become important. In the proposed approach, we first generate a feature space with kLog that captures the intrasentential properties and relations. Hereafter, these features serve as input for a structured output support vector machine that can handle sequence tagging

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 579–589, Jeju Island, Korea, 12–14 July 2012. ©2012 Association for Computational Linguistics

(Tsochantaridis et al., 2004), in order to take the intersentential features into account. Second, since there are more than two categories, and each sentence can have multiple labels, the problem is now a multiclass multilabel classification task.

The main contribution of this paper is that we show that kLog's relational nature and its ability to declaratively specify and use background knowledge is beneficial for natural language learning problems. This is shown on the NICTA-PIBOSO corpus, for which we present results that indicate a clear improvement on the state-of-the-art.

The remainder of this paper is organized as follows. In Section 2, we outline earlier work that is related to the research presented here. Section 3 describes the methodology of our method. We present a thorough evaluation of our method in Section 4. The last section draws conclusions and presents some ideas for future work.

2 Related Work

EBM is an approach to clinical problem-solving based on "systematically finding, appraising, and using contemporaneous research findings as the basis for clinical decisions" (Rosenberg and Donald, 1995). The evidence-based process consists of four steps: (1) Formulating a question from a patient's problem; (2) Searching the literature for relevant clinical articles; (3) Evaluating the evidence; And (4) implementing useful findings in clinical practice. Given the amounts of medical publications available in databases such as PubMed, automating step 2 is crucial to help doctors in their practice. Efforts in this direction from the NLP community have so far focused on corpus annotation (Demner-Fushman and Lin, 2007; Kim et al., 2011), text categorization (Davis-Desmond and Mollá, 2012), summarization (Mollá and Santiago-Martínez, 2011), and question-anwering (Niuet al., 2003; Demner-Fushman and Lin, 2007).

The existing corpora are usually annotated with the PICO mnemonic (Armstrong, 1999) concepts, that are used to build queries when searching for literature for EBM purposes. The PICO concepts are: primary Problem (P) or population, main Intervention (I), main intervention Comparison (C), and Outcome of intervention (O). PICO helps determining what terms are important in a query and therefore it helps building the query, which is sent to the search repositories. Once the documents are found, they need to be read by a person who eliminates irrelevant documents.

The first attempt to classify PICO concepts is presented in Demner-Fushman and Lin (2007), who apply a rule-based approach to identify sentences where PICO concepts occur and a supervised approach to classify sentences that contain an *Outcome*. The features used by this classifier are n-grams, position, and semantic information from the parser used to process the data. The system is trained on 275 abstracts manually annotated. The accuracies reported range from 80% for *Population*, 86% for *Problem*, 80% for *Intervention*, and, from 64% to 95% for *Outcome* depending on the test set of abstracts.

Kim et al. (2011) perform a similar classification task in two steps. First a classifier identifies the sentences that contain PICO concepts, and then another classifier assigns PICO tags to the sentences found to be relevant by the previous classifier. The system is based on a CRF algorithm and is trained on the NICTA-PIBOSO corpus. This dataset contains 1,000 medical abstracts manually annotated with an extension of the PICO tagset, for which the definitions are listed in Table 1. The annotation is performed at sentence level and one sentence may have more than one tag. An example of an annotated abstract from the corpus can be found in the supplementary material. The features used by the algorithm include features derived from the context, semantic relations, structure and sequencing of the text. The system is evaluated for 5-way and 6-way classification and results are provided apart from structured and unstructured abstracts. The F-scores for structured abstracts is 89.32% for 5-way classification and 80.88% for 6-way classification, whereas for unstructured abstracts it is 71.54% for 5-way classification and 64.66% for 6-way classification.

Chung (2009) uses CRF to classify PICO concepts by combining them with general categories associated with rhetorical roles: *Aim*, *Method*, *Results* and *Conclusion*. Her system is tested on corpora of abstracts of randomized control trials. First structured abstracts with headings labeled with PICO

Background	Material that informs and may place the current study in perspective, e.g. work that preceded the current; information about disease prevalence; etc.
Population	The group of individual persons, objects or items comprising the study's sample, or from which the sample was taken for statistical measurement
Intervention	The act of interfering with a condition to modify it or with a process to change its course (includes prevention)
Outcome	The sentence(s) that best summarizes the consequences of an intervention
Study Design	The type of study that is described in the abstract
Other	Any sentence not falling into one of the other categories and presumed to provide little help with clinical decision making, i.e. non-key or irrelevant sentences

Table 1: Definitions of the semantic tags used as annotation categories (taken from Kim et al. (2011)).

concepts are used. A sentence level classification task is performed, assigning only one rhetorical role per sentence. The F-scores obtained range from 0.93 to 0.98. Then another sentence level classification task is performed to automatically assign the labels *Intervention*, *Participant* and *Outcome Measures* to sentences in unstructured and structured abstracts without headings. F-scores of up to 0.83 and 0.84 are obtained for *Intervention* and *Outcome Measure* sentences.

Other work aimed at identifying rhetorical zones in biomedical articles. In this case areas of text are classified in terms of the rhetorical categories *Introduction*, *Methods*, *Results* and *Discussion* (IMRAD) (Agarwal and Yu, 2009) or richer categories, such as problem-setting or insight (Mizuta et al., 2006).

There exists a wide range of statistical relational learning systems (Getoor and Taskar, 2007; De Raedt et al., 2008), and many of these systems are in principle useful for natural language processing. The most popular formalism today is Markov Logic, which has already been used for natural language processing tasks such as semantic role labeling (Riedel and Meza-Ruiz, 2008) and coreference resolution (Poon and Domingos, 2008). With respect to Markov Logic, two distinguishing features of kLog are that 1) it employs kernel based methods grounded in statistical learning theory, and 2) it employs a Prolog like language for defining and using background knowledge. As Prolog is a programming language, this is more flexible that the formalism used by Markov Logic.

3 Methodology

In learning from examples, or *interpretations* (De Raedt et al., 2008), the instances are sampled identically and independently from some unknown but fixed distribution. They can be represented as pairs $z = (x, y)$, in which x represents the inputs and y the outputs. An example interpretation can be found in Figure 3, where the hasCategory relation represents y in this case, since it is the target relation we want to predict. The inputs x are formed by all other facts. The task is now to learn a function $h : X \rightarrow Y$ that maps the inputs to the outputs. Sentences may have multiple labels. Hence this is a structured output task where the output is a sequence of sets of labels attached to the sentences in a given document.

kLog is the new statistical relational language for learning with kernels that we use to tackle the PICO categories classification task. The novelty of kLog is that, based on the regular, linguistic features, it allows to define an extended high-dimensional feature space that is also able to take relational features into account in a principled manner. Furthermore, its declarative approach offers a flexible and interpretable way to construct features.

The choice of kLog is motivated by our previous results (Verbeke et al., 2012), where we showed that the relational representation of the domain as used by kLog is able to take the contextual aspects of language into account. Whereas there we only used the relations at the sentence level, the current task adds a new level of complexity, since the identification of PICO categories in abstracts also requires to take into account various relations between the sentences of an abstract. The general workflow of our approach is depicted in Figure 1, which will be de-

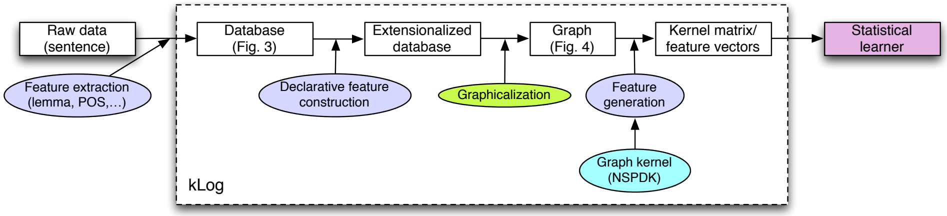

Figure 1: General kLog workflow.

scribed step by step in the following paragraphs.

Preprocessing The sentences have been preprocessed with a named entity tagger and a dependency parser.

Named entity tagging has been performed with the BiogaphTA named entity module, which matches token sequences with entries in the UMLS database[1]. UMLS integrates over 2 million names for some 900,000 concepts from more than 60 families of biomedical vocabularies (Bodenreider, 2004). The tagger matches sequences with a length of maximum 4 tokens. This covers 66.2% of the UMLS entries. By using UMLS, different token sequences referring to the same concept can be mapped to the same concept identifier (CID). The BiographTA named entity tagger has been evaluated on the BioInfer corpus (Pyysalo et al., 2007) obtaining a 72.02 F1 score.

Dependency parsing has been performed with the GENIA dependency parser GDep (Sagae and Tsujii, 2007), which uses a best-first probabilistic shift-reduce algorithm based on the LR algorithm (Knuth, 1965) and extended by the pseudo-projective parsing technique. This parser is a version of the KSDep dependency parser trained on the GENIA Treebank for parsing biomedical text. KSDep was evaluated in the CoNLL Shared Task 2007 obtaining a Labeled Attachment Score of 89.01% for the English dataset. GDEP outputs the lemmas, chunks, Genia named entities and dependency relations of the tokens in a sentence.

This information can be represented as an Entity/Relationship (E/R) diagram, a modeling paradigm that is frequently used in database theory (Garcia-Molina et al., 2008). The E/R-model for the problem under consideration is shown in Figure 2, which provides an abstract representation of the examples, i.e. medical abstracts in this case. We will show later how this abstract representation can be unrolled for each example, resulting in a graph; cf. also Figure 4 for our example sentence. This relational database representation will serve as the input for kLog.

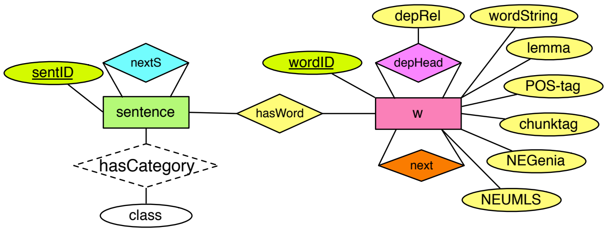

Figure 2: E/R-diagram modeling the sentence identification task.

The *entities* are the words and sentences in the abstract. They are represented by the rectangles in the E/R-model. Each entity can have a number of properties attached to it, depicted by the ovals and has a unique identifier (underlined properties). As in database theory, each entity corresponds with a tuple, or *fact*, in the database.

Figure 3 shows a part of an example interpretation z. For example, w(w4_1,‘Surgical’,‘Surgical’,b-np,jj,‘O’,‘O’) specifies a word entity, with w4_1 as identifier and the other arguments as properties. As indicated before, as lexical information we take the token string itself, its lemma, the part-of-speech tag and the chunk tag into account. We also include some semantic information, namely two binary values indicating if the word is a (biological) named entity. sentence(s4,4) represents a sentence entity, with its index in the abstract as a property.

Furthermore, the E/R-diagram also contains a number of *relationships*, which are represented by

[1] From UMLS only the MRCONSO.RRF and MRSTY.RRF files are used.

```
sentence(s4,4)
hasCategory(s4,‘background’)
w(w4_1,‘Surgical’,‘Surgical’,b-np,
jj,‘O’,‘O’) hasWord(s4,w4_1)
dh(w4_1,w4_2,nmod)
nextW(w4_2,w4_1)
w(w4_2,‘excision’,‘excision’,i-np,
nn,‘O’,‘O’) hasWord(s4,w4_2)
dh(w4_2,w4_5,sub)
nextW(w4_3,w4_2)
w(w4_3,‘of’,‘of’,b-pp,in,’O’,’O’)
hasWord(s4,w4_3)
dh(w4_3,w4_2,nmod)
nextW(w4_4,w4_3)
w(w4_4,‘CNV’,‘CNV’,b-np,nn,
‘B-protein’,‘O’) hasWord(s4,w4_4)
dh(w4_4,w4_3,pmod)
nextW(w4_5,w4_4)
...
```

Figure 3: Part of an example interpretation z, representing the example sentence in Figure 4.

the diamonds. They are linked to the entities that participate in the relationship, or stand alone if they characterize general properties of the interpretation. An example relation is nextW(w4_2,w4_1), which indicates the sequence of the words in the sentence. dh(w4_1,w4_2,nmod) specifies that word w4_1 is a noun modifier of word w4_2, and thus serves to incorporate the dependency relationships between the words. hasCategory(s4,‘background’) signifies that sentence s4 is a sentence describing background information. This relation is the target relation that we want to predict for this task and will not be taken into account as a feature, but is listed in the database and only used during the training of the model.

Since the previously described entities and relationships are listed explicitly in the database, these are called *extensional relations*, in contrast to the *intensional relations*, as we will describe next.

Declarative feature construction A strength of kLog is that it is also capable of constructing features *declaratively*, by using intensional relations. This enables one to encode additional background knowledge based on a small set of preprocessed features, which renders experimentation very flexible and makes the results more interpretable. It furthermore allows one to limit the required features to the core discriminative ones. These intensional features are defined through definite clauses, and is done using an extension of the declarative programming language Prolog. The following features were used. We make a distinction between the features used for structured and unstructured abstracts.

For structured abstracts, four intensional relations were defined. The relation lemmaRoot(S,L) is specified as:

```
lemmaRoot(S,L) ←
   hasWord(S, I),
   w(I,_,L,_,_,_,_),
   dh(I,_,root).
```

For each sentence, it only selects the lemmas of the root word in the dependency tree, which markedly limits the number of word features used. The following relations are related to, and try to capture the document structure imposed by the section headers present in the structured abstracts. hasHeaderWord(S,X) identifies whether a sentence is a header of a section. In order to realize this, it selects the words of a sentence that count more than four characters (to discard short names of biological entities), which all need to be uppercase.

```
hasHeaderWord(S,X) ←
   w(W,X,_,_,_,_,_),
   hasWord(S,W),
   (atom(X) -> name(X,C) ; C = X),
   length(C,Len),
   Len > 4,
   all_upper(C).
```

Also the sentences below a certain section header need to be marked as belonging to this section, which is done by the relation hasSectionHeader(S,X).

```
hasSectionHeader(S,X) ←
   nextS(S1,S),
   hasHeaderWord(S1,X).
hasSectionHeader(S,X) ←
   nextS(S1,S),
   not isHeaderSentence(S),
   once(hasSectionHeader(S1,X)).
```

For the unstructured abstracts, also the lemmaRoot relation is used, but next to the lemma, now also the part-of-speech tag of the root word is taken into account. Since the unstructured abstracts lack section headers, other features were needed to distinguish between the different sections, for which the relation prevLemmaRoot proved to be very informative. It adds the lemma of the root word in the previous sentence as a property to the current sentence under consideration.

```
prevLemmaRoot(S,L) ←
    nextS(S1,S),
    lemmaRoot(S1,L,_).
```

The intensional predicates are grounded. This is a proces similar to materialization in databases, that is, the atoms implied by the background knowledge and the facts in the example are all computed using Prolog's deduction mechanism. This leads to the *extensionalized database*, in which both the extensional as well as the grounded intensional predicates are listed.

Graphicalization and feature generation In the third step, the interpretations are *graphicalized*, i.e. transformed into graphs. Since the facts that form the interpretation still conform to the E/R-diagram, this can be interpreted as unfolding the E/R-diagram over the data. An example illustrating this process is given in Figure 4. Each interpretation is converted into a bipartite graph, for which there is a vertex for every ground atom of every E-relation, one for every ground atom of every R-relation, and an undirected edge $\{e, r\}$ if an entity e participates in relationship r.

The obtained graphs can then be used in the next step for *feature generation*. This is done by means of a graph kernel κ, which calculates the similarity between two graphicalized interpretations. Any graph kernel that allows fast computations on large graphs and has a flexible bias to enable heterogeneous features can in theory be applied. In the current implementation, an extension of the Neighborhood Subgraph Pairwise Distance Kernel (NSPDK) (Costa and De Grave, 2010) is used.

NSPDK is a decomposition kernel (Haussler, 1999), in which pairs of subgraphs are compared to each other in order to calculate the similarity between two graphs. These subgraphs can be seen as circles in the graph, and are defined by three hyperparameters. First of all, there is the center of the subgraph, the *kernel point*, which can be any entity or relation in the graph. The entities and relations to be taken into account as kernel points are marked beforehand as a subset of the intensional and extensional domain relations. The *radius* r determines the size of the subgraphs and defines which entities or relations around the kernel point are taken into account. Each entity or relation that is within a number of r edges away from the kernel point is considered to be part of the subgraph. The third hyperparameter, the *distance* d, determines how far apart from each other the kernel points can be. Each subgraph around a kernel point that is within a distance d or less from the current kernel point will be considered. This is captured by the relation $R_{r,d}(A_v, B_u, G)$ between two rooted subgraphs A_v, B_u and a graph G, which selects all pairs of neighborhood graphs of radius r whose roots are at distance d in a given graph G.

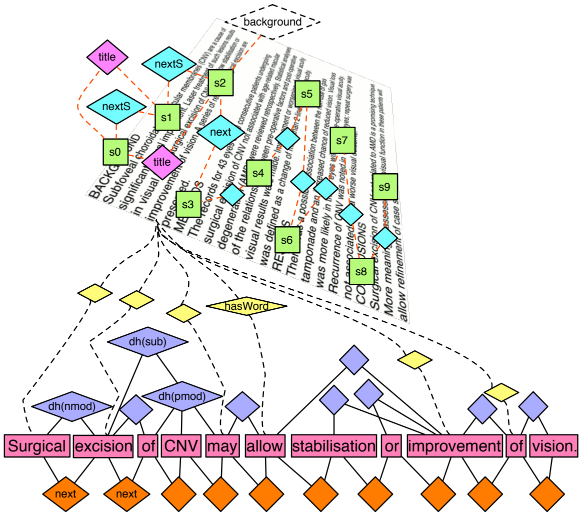

Figure 4: Graphicalization G_z of interpretation z.

The kernel $\kappa_{r,d}(G, G')$ between graphs G and G' on the relation $R_{r,d}$ is then defined as:

$$\kappa_{r,d}(G, G') = \sum_{\substack{A_v, B_u \in R^{-1}_{r,d}(G) \\ A'_{v'}, B'_{u'} \in R^{-1}_{r,d}(G')}} \delta(A_v, A'_{v'})\delta(B_u, B'_{u'}) \quad (1)$$

For efficiency reasons, an upper bound is imposed on the radius and distance parameters, which leads to the following kernel definition:

$$K_{r^*,d^*}(G, G') = \sum_{r=0}^{r^*} \sum_{d=0}^{d^*} \kappa_{r,d}(G, G') \qquad (2)$$

We hereby limit the sum of the $\kappa_{r,d}$ kernels for all increasing values of the radius and distance parameter up to a maximum given value of r^*, respectively d^*.

The result of this graphicalization and feature generation process is an extended, high-dimensional feature space, which serves as input for the statistical learner in the next step.

Learning The constructed feature space contains one feature vector per sentence. This implies that the sequence information of the sentences at the document level is not taken into account yet. Since the order of the sentences in the abstract is a valuable feature for this prediction problem, a learner that reflects this in the learning process is needed, although in principle any statistical learner can be used on the feature space constructed by kLog. Therefore we opted for SVM-HMM[2] (Tsochantaridis et al., 2004), which is an implementation of structural support vector machines for sequence tagging. In contrast to a conventional Hidden Markov Model, SVM-HMM is able to take these entire feature vectors as observations, and not just atomic tokens.

In our case, the instances to be tagged are formed by the sentences for which feature vectors were created in the previous step. The *qid* is a special feature that is used in the structured SVM to restrict the generation of constraints. Since every document needs to be represented as a sequence of sentences, in SVM-HMM, the qid's are used to obtain the document structure. The order of the HMM was set to 2, which means that the two previous sentences were considered for collective classification. The cost value was set to 500, and was determined via cross-validation. For epsilon, the default value, 0.5, was kept, since this mainly only influences the running time and memory consumption during training.

[2] http://www.cs.cornell.edu/people/tj/svm_light/svm_hmm.html

	All	**S**	**U**
Nb. Abstracts	1000	376	624
Nb. Sentences	10379	4774	5605
- Background	2557	669	1888
- Intervention	690	313	377
- Outcome	4523	2240	2283
- Population	812	369	443
- Study Design	233	149	84
- Other	1564	1034	530

Table 2: Number of abstracts and sentences for Structured (S) and Unstructured (U) abstract sets, including number of sentences per class (taken from (Kim et al., 2011)).

4 Evaluation

We evaluate the performance of kLog against a baseline system and a memory-based tagger (Daelemans and van den Bosch, 2005). The results are also compared against those from Kim et al. (2011), which is the state-of-the-art system for this task.

4.1 Datasets

We perform our experiments on the NICTA-PIBOSO dataset from Kim et al. (2011) (kindly provided by the authors). It contains 1,000 abstracts of which 500 were retrieved from MEDLINE by querying for diverse aspects in the traumatic brain injury and spinal cord injury domain. The dataset consists of two types of abstracts. If the abstract contains section headings (e.g. *Background*, *Methodology*, *Results*, etc.), it is considered to be *structured.* This information can be used as a feature in the model. The other abstracts are regarded *unstructured.*

The definitions of the semantic tags used as annotations categories are a variation on the PICO tag set, with the addition of two additional categories (see Table 1 in Section 2). Each sentence can be annotated with multiple classes. This renders the task a multiclass multilabel classification problem. The statistics on this dataset can be found in Table 2.

In order to apply the same evaluation setting as Kim et al. (2011), we used the dataset from Demner-Fushman et al. (2005) as external dataset. It consists of 100 sentences of which 51 are structured. Because the semantic tag set used for annotation slightly differs from the one presented in Table 1, and to make our results comparable, we will use the same mapping as used in Kim et al. (2011).

4.2 Baseline and benchmarks

We compare the kLog system to three other systems: a baseline system, a memory-based system, and the scores reported by Kim et al. (2011).

The memory-based system that we use is based on the memory-based tagger MBT[3] (Daelemans and van den Bosch, 2005). This machine learner is originally designed for part-of-speech tagging. It processes data on a sentence basis by carrying out sequential tagging, *viz.* the class label or other features from previously tagged tokens can be used when classifying a new token. In our setup, the sentences of an abstract are taken as the processing unit and the collection of all sentences in an abstract is taken as one sequence.

The features that are used to label a sentence are the class labels of four previous sentences, the ambitags of the following two sentences, the lemma of the dependency root of the sentence, the position of the sentence in the abstract, the lemma of the root of the previous sentence, and section information. For each root lemma, all possible class labels, as observed in the training data, are concatenated into one ambitag. These tags are stored in a list. An ambitag for a sentence is retrieved by looking up the root lemma in this list. The position of the sentence is expressed by a number. Section information is obtained by looking for a previous sentence that consists of only one token in uppercase. Finally, basic lemmatization is carried out by removing a final *S*. All other settings of MBT are the default settings and no feature optimization nor feature selection has been carried out to prevent overfitting.

When a class label contains multiple labels, like e.g. *population* and *study design*, these labels are concatenated in an alphabetically sorted manner. This method of working reduces the multilabel problem to a problem with many different labels, i.e. the label powerset method of Tsoumakas et al. (2010).

The baseline system is exactly the same as the memory-based system except that no machine learner is included. The most frequent class label in the training data, i.e. *Outcome*, is assigned to each instance. The memory-based system enables us to compare kLog against a basic machine learning approach, using few features. The majority baseline system enables us to compare the memory-based system and kLog against a baseline in which no information about the observations is used.

4.3 Parametrization

From the kernel definition it might be clear that the kLog hyperparameters, namely the distance d and radius r, can have a strong influence on the results. This requires a deliberate choice during parametrization. From a linguistic perspective, the use of unigrams and bigrams is justifiable, since most phrases that reveal clues on the structure of the abstract (e.g. *evaluation measures*, *methodolody*, *future work*) can be expressed with single or pairs of words. This is reflected by a distance and radius both set to 1, which enables to take all possible combinations of consecutive words into account and captures the relational information attached to the word in focus, i.e. the current kernel point. This is confirmed by cross-validation on other settings for the hyperparameters.

Since kLog generates a feature vector, only the sequence information at word level is taken into account by kLog. Since we use a sequence labeling approach as statistical learner, i.e. SVM-HMM, at the level of the abstract this information is however implicitly taken into account during learning. For SVM-HMM, only the cost parameter C, which regulates the trade-off between the slack and the magnitude of the weight-vector, and ϵ, that specifies the precision to which constraints are required to be satisfied by the solution, were optimized by means of cross-validation. For the other parameters, the default values were used.

4.4 Results

Experiments are run on structured and unstructured abstracts separately. On the NICTA-PIBOSO corpus, we performed 10-fold cross-validation. Over all folds, all labels, i.e. the parts of the multilabels, are compared in a binary way between gold standard and prediction. Summing all true positives, false positives, and false negatives over all folds leads to micro-averaged F-scores. This was done for two different settings. In one setting, *CV/6-way*, we combined the labeling of the sentences with the identification of irrelevant information, by adding the *Other*

[3] `http://ilk.uvt.nl/mbt` [16 March 2012]

label as an extra class in the classification. The results are listed in Table 3.

CV/6-way	MBT		Kim et al.		kLog	
Label	**S**	**U**	**S**	**U**	**S**	**U**
Background	71.0	61.3	81.84	68.46	86.19	76.90
Intervention	24.3	6.4	20.25	12.68	26.05	16.14
Outcome	87.9	70.4	92.32	72.94	92.99	77.69
Population	50.6	15.9	56.25	39.80	35.62	21.58
Study Design	45.9	13.10	43.95	4.40	45.5	6.67
Other	86.1	20.9	69.98	24.28	87.98	24.42

Table 3: F-scores per class for structured (S) and unstructured (U) abstracts.

For this setting, kLog is able to outperform both MBT and the system of Kim et al. (2011), for both structured and unstructured abstracts on all classes except *Population*. From Table 4, where the micro-average F-scores over all classes and for all settings are listed, it can be observed that kLog performs up to 3.73% better than MBT over structured abstracts, and 9.67% better over unstructured ones.

Although to a lesser extent for the structured abstracts, the same pattern can be observed for the *CV/5-way* setting, where we tried to classify the sentences only, without considering the irrelevant ones. The per-class results for this setting are shown in Table 5. Now the scores for *Population* are comparable to the other systems, due to which we assume these sentences are similar in structure to the ones labeled with *Other*.

For the external corpus, the results are listed in Table 6. Although kLog performs comparably for the individual classes *Background* and *Intervention*, its overall performance is worse on the structured abstracts. In case of the unstructured abstracts, kLog performs better on the majority of the individual classes and in overall performance for the 5-way setting, and comparable for the 4-way setting.

	Baseline		MBT		kLog	
Method	**S**	**U**	**S**	**U**	**S**	**U**
CV/6-way	43.90	41.87	80.56	57.47	84.29	67.14
CV/5-way	61.79	46.66	86.96	64.37	87.67	72.95
Ext/5-way	66.18	6.76	36.34	11.56	20.50	14.00
Ext/4-way	30.11	27.23	67.29	55.96	50.40	50.50

Table 4: Micro-averaged F1-score obtained for structured (S) and unstructured (U) abstracts, both for 10-fold cross-validation (CV) and on the external corpus (Ext).

CV/5-way	MBT		Kim et al.		kLog	
Label	**S**	**U**	**S**	**U**	**S**	**U**
Background	87.1	64.9	87.92	70.67	91.45	80.06
Intervention	48.0	6.9	48.08	21.39	45.58	22.65
Outcome	95.8	75.9	96.03	80.51	96.21	83.04
Population	70.9	21.4	63.88	43.15	63.96	23.32
Study Design	50.0	7.4	47.44	8.6	48.08	4.50

Table 5: F-scores per class for 5-way classification over structured (S) and unstructured (U) abstracts.

	MBT		Kim et al.		kLog	
Label	**S**	**U**	**S**	**U**	**S**	**U**
			Ext/5-way			
Background	58.9	15.7	56.18	15.67	58.30	29.10
Intervention	21.5	13.8	15.38	28.57	40.00	34.30
Outcome	29.3	17.8	81.34	60.45	27.80	24.10
Population	10.7	17.8	35.62	28.07	5.60	28.60
Other	40.7	3.5	46.32	15.77	11.40	8.50
			Ext/4-way			
Background	90.4	67.5	77.27	37.5	65	68.6
Intervention	29	23.1	28.17	8.33	28.1	32.3
Outcome	74.1	74.6	90.5	78.77	72.4	72.7
Population	48.7	23.8	42.86	28.57	11.8	15.4

Table 6: F-scores per class for 5-way and 4-way classification over structured (S) and unstructured (U) abstracts on the external corpus.

As a general observation, it is important to note that there is a high variability between the different labels. Due to kLog's ability to take the structured input into account, we assume a correlation between the sentence structure of the label and the prediction quality. We intend to perform an extensive error analysis, in order to detect patterns which may allow us to incorporate additional declarative background knowledge into our model.

5 Conclusions

We presented a statistical relational learning approach for the automatic identification of PICO categories in medical abstracts. To this extent, we used kLog, a new framework for logical and relational learning with kernels. Due to its graphical approach, it is able to exploit the full relational representation, that is often inherent in language structure. Since contextual features are often essential and relations are prevalent, the aim of this paper was to show that statistical relational learning in general, and the graph kernel-based approach of kLog in particular, is specifically suited for problems in natural language learning.

In future work, we intend to explore additional ways to incorporate background knowledge in a declarative way, since it renders the language learning problem more intuitive and gives a better understanding of feature contribution. Furthermore, we also want to investigate the use of SRL approaches for high-relational domains, and make a clear comparison with related techniques.

6 Acknowledgements

This research is funded by the Research Foundation Flanders (FWO project G.0478.10 - Statistical Relational Learning of Natural Language), and made possible through financial support from the KU Leuven Research Fund (GOA project 2008/08 Probabilistic Logic Learning), the University of Antwerp (GOA project BIOGRAPH) and the Italian Ministry of Education, University, and Research (PRIN project 2009LNP494 - Statistical Relational Learning: Algorithms and Applications). The authors would like to thank Fabrizio Costa, Kurt De Grave and the anonymous reviewers for their valuable feedback.

References

Shashank Agarwal and Hong Yu. 2009. Automatically Classifying Sentences in Full-text Biomedical Articles into Introduction, Methods, Results and Discussion. *Bioinformatics*, 25(23):3174–3180.

E. C. Armstrong. 1999. The Well-built Clinical Question: the Key to Finding the Best Evidence Efficiently. *WMJ*, 98(2):25–28.

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): Integrating Biomedical Terminology. *Nucleic Acids Research*, 32(Suppl.1):D267–D270.

Grace Y Chung. 2009. Sentence Retrieval for Abstracts of Randomized Controlled Trials. *BMC Medical Informatics and Decision Making*, 9(10).

Fabrizio Costa and Kurt De Grave. 2010. Fast Neighborhood Subgraph Pairwise Distance Kernel. *Proceedings of the 26th International Conference on Machine Learning*, 255–262, Haifa, Israel. Omnipress.

Walter Daelemans and Antal van den Bosch. 2005. *Memory-Based Language Processing.*. Studies in Natural Language Processing. Cambridge University Press, Cambridge, UK.

P. Davis-Desmond and Diego Mollá. 2012. Detection of Evidence in Clinical Research Papers. *Proceedings of the Australasian Workshop On Health Informatics and Knowledge Management (HIKM 2012)*, Melbourne, Australia, 129:13–20. Australian Computer Society, Inc.

Dina Demner-Fushman, Barbara Few, Susan E. Hauser, and George Thoma. 2005. Automatically Identifying Health Outcome Information in MEDLINE Records. *Journal of the American Medical Informatics Association (JAMIA)*, 13:52–60.

Dina Demner-Fushman and Jimmy Lin. 2007. Answering Clinical Questions with Knowledge Based and Statistical Techniques. *Computational Linguistics*, 33(1):63–103.

Luc De Raedt, Paolo Frasconi, Kristian Kersting, and Stephen Muggleton, editors. 2008. *Probabilistic Inductive Logic Programming*. In: Lecture Notes in Computer Science (LNCS), 4911. Springer-Verlag, Heidelberg, Germany.

Paolo Frasconi, Fabrizio Costa, Luc De Raedt, and Kurt De Grave. 2012. kLog - a Language for Logical and Relational Learning with Kernels. *arXiv:1205.3981v2*.

Hector Garcia-Molina, Jeff Ullman, and Jennifer Widom. 2008. *Database Systems: The Complete Book*. Prentice Hall Press, Englewood Cliffs, NJ, USA.

Lise Getoor and Ben Taskar. 2007. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, Cambridge, MA, USA.

David Haussler. 1999. Convolution kernels on discrete structures. *Technical report (UCSC-CRL-99-10), University of California at Santa Cruz.*

Su Nam Kim, David Martinez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic Classification of Sentences to Support Evidence Based Medicine. *BMC Bioinformatics*, 12(2):S5.

Donald E. Knuth. 1965. On the Translation of Languages from Left to Right. *Information and Control*, 8: 607–639.

Y. Mizuta, A. Korhonen, T. Mullen, and N. Collier. 2006. Zone Analysis in Biology Articles as a Basis for Information Extraction. *International Journal of Medical Informatics*, 75(6):468–487.

Diego Mollá and Mara Elena Santiago-Martínez. 2011. Development of a Corpus for Evidence Medicine Summarisation. *Proceedings of the 2011 Australasian Language Technology Workshop (ALTA 2011)*, Canberra, Australia, 86–94. Association for Computational Linguistics.

Yun Niu, Graeme Hirst, Gregory McArthur, and Patricia Rodriguez-Gianolli. 2003. Answering Clinical

Questions with Role Identification. *Proceedings of the ACL, Workshop on Natural Language Processing in Biomedicine*. Sapporo, Japan, 73–80. Association for Computational Linguistics.

Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with Markov logic. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008*. Honolulu, Hawaii, 650–659. Association for Computational Linguistics.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. 2007. BioInfer: a Corpus for Information Extraction in the Biomedical Domain. *BMC Bioinformatics*, 8:50.

Sebastian Riedel and Ivan Meza-Ruiz. 2008. Collective semantic role labelling with Markov logic. *Proceedings of the Twelfth Conference on Computational Natural Language Learning (CoNLL 2008*. Manchester, United Kingdom, 193–197. Association for Computational Linguistics.

William Rosenberg and Anna Donald. 1995. Evidence Based Medicine: an Approach to Clinical Problem Solving. *British Medical Journal*, 310(6987):1122–1126.

Kenji Sagae and Jun'ichi Tsujii. 2007. Dependency Parsing and Domain Adaptation with LR Models and Parser Ensembles. *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, Prague, Czech Republic, 1044–1050. Association for Computational Linguistics.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support Vector Machine Learning for Interdependent and Structured Output Spaces. *Proceedings of the twenty-first international conference on Machine learning (ICML)*, Alberta, Canada, 104–111. ACM.

Grigorios Tsoumakas and Ioannis Katakis and Ioannis P. Vlahavas. Oded Maimon and Lior Rokach, editors. 2010. Mining Multi-label Data. In: *Data Mining and Knowledge Discovery Handbook*, 2nd ed., 667–685. Springer-Verlag, Heidelberg, Germany.

Mathias Verbeke, Paolo Frasconi, Vincent Van Asch, Roser Morante, Walter Daelemans, and Luc De Raedt. 2012. Kernel-based Logical and Relational Learning with kLog for Hedge Cue Detection. *Proceedings of the 21th International Conference on Inductive Logic Programming*, in press.