Combining Bayesian and Support Vector Machines Learning to automatically complete Syntactical Information for HPSG-like Formalisms

Manolis Maragoudakis, Katia Kermanidis, Nikos Fakotakis, George Kokkinakis

Wire Communications Laboratory
Department of Electrical and Computer Engineering
University of Patras
26500 Rion, Patras, Greece
{mmarag,kerman,fakotaki,gkokkin}@wcl.ee.upatras.gr

Abstract

Learning Bayesian Belief Networks (BBN) from corpora and incorporating the extracted inferring knowledge with a Support Vector Machines (SVM) classifier has been applied to the automatic acquisition of verb subcategorization frames for Modern Greek. We have made use of minimal linguistic resources, such as basic morphological tagging and phrase chunking, to demonstrate that verb subcategorization, which is of great significance for developing robust natural language human computer interaction systems, could be achieved using large corpora, without having any general-purpose syntactic parser at all. Moreover, by taking advantage of the plethora in unlabeled data found in text corpora in addition to some available labeled examples, we overcome the expensive task of annotating the whole set of training data and the performance of the subcategorization frames learner is increased. We argue that a classifier generated from BBN and SVM is well suited for learning to identify verb subcategorization frames. Empirical results will support this claim. Performance has been methodically evaluated using two different corpora, one balanced and one domain-specific in order to determine the unbiased behavior of the trained models. Limited training data are proved to endow with satisfactory results. We have been able to achieve precision exceeding 90% on the identification of subcategorization frames which were not known beforehand. The obtained valid frames have been used to fill out the subcategorization field of verb entries in an HPSG-like lexicon using the LKB grammar development environment.

1. Introduction

Verb subcategorization is an important issue especially for parsing and grammar development as it provides the parser with syntactic and/or semantic information on a verb's arguments, that is the set of restrictions the verb imposes on its arguments. In many natural language interface applications, the syntactic-semantic information extracted from subcategorization frames (SF) could prove to be essential since it often clarifies the agent and the receiver of an action.

Verb subcategorization information is essential for the creation of HPSG-like lexicons. Handcrafting of such lexicons so as to implement an HPSG grammar that allows for a uniform and systematic description of a language. Filling the lexicon automatically with syntactic subcategorization information is an important step towards the completion of such a rich linguistic resource.

Nowadays, with the impressive increase in the number of available text corpora and language resources in general, the need for fully annotated syntactic parsers could be alleviated by mining subcategorization information from large text corpora. Machine-readable dictionaries listing SF usually provide only expected frames rather than actual ones and are therefore incomplete or in many cases unavailable for some languages, including Modern Greek. Considering also that building verb subcategorization classifiers is difficult and time-consuming, learning classifiers from examples is advantageous.

Previous work on learning SF focuses mainly on English (Brent, 1993; Manning, 1993; Briscoe and Carroll, 1998; Gahl, 1998). Basili et al. (1997) deal with Italian, De Lima (1997) and Eckle and Heid (1996) with German, Kawahara et al. (2000) with Japanese, Sarkar 2000) with Czech. In the earlier of these approaches only a small number of frames are learned (Brent, 1993, Manning, 1993). In most of them, the entire set of frames is known beforehand and the training text used is usually fully parsed.

The work presented in this paper is differentiated from previous research in three directions. Given that a wide-coverage syntactic parser for Modern Greek is not yet available, we first employ a set of robust pre-processing techniques that reach the stage of basic shallow parsing for obtaining the necessary grammatical and syntactic information for the task at hand. Acquiring SF information from large corpora can be considered as a shallow parsing task since only key parts of the syntactic structure of a verb rather than detailed syntactic or semantic analysis is extracted.

The need for annotated training data imposed by supervised learning methods cannot always be easily met. Especially in the task of detecting the arguments of a verb, which is not straightforward even for linguists, class labeling of the whole set of training data by hand can be very difficult as well as expensive. For that reason, as a second novelty, we present an approach that augments the given labeled training set with unlabeled instances in order to achieve classification improvement. More specifically, our method is based on the probabilistic analysis of support vector machines (SVM). Previous research on this topic (Sollich, 2001, Gao et al., 2001, Cristianini et al., 1998) has revealed the existence of a probabilistic interpretation of SVM concerning the kernel hyperparameters tuning. We are attempting to estimate the prior distribution of the latter using Bayesian belief network (BBN) learning theory. A more detailed analysis of the proposed framework is discussed in following sections. Again, experimental simulations demonstrate that the use of unlabeled data reduces classification error by up to 6%. In our work, by taking advantage of large-size unlabeled corpora in addition to some available labeled examples, we overcome the expensive task of annotating the whole set of training data and we increase the performance of the learner compared to the performance after training with the set of a restricted number of labeled instances.

Finally, the complete set of frames for a particular verb is not known to us beforehand. It is learned automatically through the training process. It is more probable for subsets of a frame to be encountered within the environment of a verb, while the complete subcategorization information of the latter is unlikely to appear in a single occurrence of the verb. We take the above observations into account when trying to estimate the relative frequencies of co-occurrence of a verb with a frame in order to establish a baseline performance metric for the task at hand.

We are incorporating two machine-learning methods that have revealed great potential for learning classification functions and have not been previously used for the detection of verb SF. We apply BBN learning from corpora and use the extracting network as an inference tool that enables automatic acquisition of SF for Modern Greek. Furthermore, we experiment with SVM, a recently well-founded technique in terms of computational learning theory that has been successfully applied in numerous classification problems including text categorization (Joachims, 1996), pattern recognition, face detection (Osuna et al., 1997) etc. Experimental results support the claim that both BBN and SVM are well-suited classifiers for the verb SF domain.

Modern Greek (MG) is a ‘semi-free’ word order language. Position of the syntactic constituents in a sentence is a very weak indicator of the syntactic role of the constituent. Moreover, the existence of adverbs in the neighborhood of a verb is a major source of noise.

2. Properties of Modern Greek

As mentioned above, MG is a ‘free phrase-order’ language. Although words within a phrase have more or less fixed positions, phrases can form a sentence in almost any ordering. Noun phrases (NP) (a), prepositional phrases (PP) (b), adverbs (c) and secondary clauses (d) may function as arguments to verbs. Weak personal pronouns (WPP) may also function as arguments to verbs (e) when occurring within the verb phrase either in the genitive or accusative case. These arguments are not determined by their position but by their morphology (especially their case), by the preposition introducing a PP, by the keyword (conjunction, adverb or pronoun) introducing a secondary clause or by the lemma in the case of an adverb. The examples in Table 1 illustrate how a single verb (πιστεύω - to believe) can take as arguments all of the above syntactic constituents. The brackets located next to a word describe the type of phrase and the case of the word.

a. Πιστεύω την Ελένη Believe Helen[NPacc] *I believe Helen*
b. Πιστεύω στο Θεό Believe in[PREP] God[acc] *I believe in God*
c. Έτσι πιστεύω So[ADV] believe *I believe so*
d. Πιστεύω πως θα έρθει Believe that[CONJ] come *I believe that he will come*
e. Σε πιστεύω You[WPPacc] believe *I believe you*

Table 1: Examples of various syntactic constituents of the verb “πιστεύω” (to believe)

3. Head driven Phrase Structure Grammar (HPSG)

HPSG is an approach to constraint-based grammatical theory that attempts to present a model of human language based on the logic of typed feature structures. Types are hierarchically ordered in an inheritance hierarchy and specify a set of features that are appropriate for them as well as the values that these features take.

Lexicalism is one of the main characteristics of HPSG, i.e. lexical entries provide a very important part of the linguistic (syntactic and semantic) information. The interaction between the information of lexical entries enables linguistic processing.

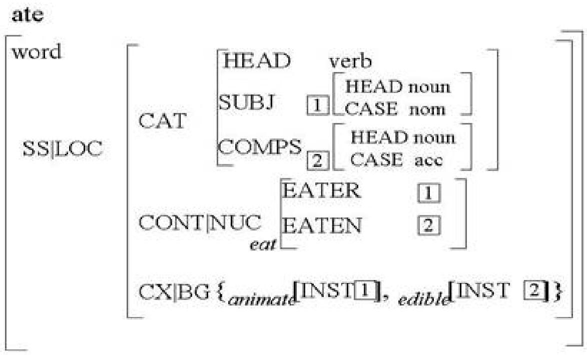
ate
word
SS|LOC
CAT
HEAD verb
SUBJ 1 HEAD noun CASE nom
COMPS 2 HEAD noun CASE acc
CONT|NUC eat
EATER 1
EATEN 2
CX|BG { animate [INST 1], edible [INST 2] }

Figure 1. A representation of the lexicon entry for “ate” in the HPSG formalism

A representation of part of a typed feature structure for the lexical entry of the verb ate is shown in figure 1. Every linguistic entity (words, phrases, sentences) in HPSG is represented by sign, a type that contains the phonological, syntactic and semantic information of the entity. The synsem (abbreviated as SS in the figure) feature of a sign contains all the syntactic and semantic information associated with that sign. The nonlocal feature (omitted in the figure) of synsem is used to represent information concerning long-distance relationships while the local feature (LOC) includes the feature category (CAT) for categorial and subcategorization information and the content (CONT) and context/background (CX|BG) feature to accommodate selectional restrictions (Androutsopoulos and Dale, 2000). For almost every phrasal structure, there is a daughter node (head-daughter) with a set of properties which are relevant at the level of the mother node. These properties constitute the head feature structure of type head (subtype of local).

The output of the combination of one or more feature structures in a particular way is specified by a handful of phrase structure schemata, the grammar rules. They are defined as general as possible and are therefore quite powerful. During the application of a schema, feature unification is handled by a set of HPSG principles that determine the validity of the output sign and ensure the well-formedness of phrases in higher tree levels. The Head Feature Principle, for example, makes sure that in every headed phrase, the value of the head feature of the mother and the value of the head feature of the head-daughter are unified. As an example of a rule, the head-complement schema takes a word and its complement and constructs a phrase that takes no complements (the value of the comps feature (COMPS) introduced by the type category will be an empty list). A complete and detailed presentation of HPSG can be found in Sag and Wasow (1999).

In the present work we are concerned with the automatic acquisition of the information required by the CAT field. In the example, ate requires a subject that is a linguistic entity with a noun head in the nominative case and a complement which is a linguistic entity with a noun head in the accusative case. It should be noted that the COMPS field is actually a list of possibly more than one element. We describe a methodology with which verb subcategorization information may be acquired directly from corpora in order to enable the construction of an HPSG-like lexicon that contains such information. The index numbers 1 and 2 in the boxes are used to denote the fact that the syntactical subject and object of ate are identical to the eater (of semantic type animate) and to what is being eaten (of semantic type edible) respectively.

4. Corpus pre-processing

Only limited linguistic resources have been exploited for the pre-processing of the corpora, given the fact that we want to bypass the need of a wide-coverage syntactic parser. More specifically, pre-processing was realized in the following stages:

- Basic morphological tagging
- Chunking
- Headword detecting for noun, i.e. the word the grammatical properties of which are inherited by the phrase
- Filtering

MG is a highly inflected language, rich in morphology. Case and voice information is essential for frame detection. In more detail, morphological tagging (Sgarbas et al., 1999) includes

- part-of-speech tagging for all words
- case tagging for nouns, adjectives and pronouns
- voice tagging for verbs
- lemmatising of conjunctions

Most words belonging to a closed-class part-of-speech category are divided into subcategories depending on their type (for example personal / relative / interrogative pronouns). We use information like this instead of the lemma of the word, to decrease the number of the tag classes used. It is important, however, not to leave out necessary knowledge by omitting the lemma. More specifically, type tagging consists of:

- type tagging for pronouns (distinguishing among relative, interrogative and the rest of the pronouns)
- type tagging for conjunctions (distinguishing between coordinate and subordinate conjunctions).

The phrase chunker (Stamatatos et al., 2000) is based on very limited linguistic resources, i.e. a small keyword lexicon containing some 450 keywords (closed-class words) and a suffix lexicon of 300 of the most common word suffixes in MG. It robustly detects the boundaries of intrasentential noun phrases, prepositional phrases, verb phrases, adverbial phrases and conjunctions.

Identifying the headword of a noun phrase, i.e. the word that holds the morphological information (case, number) of the whole phrase, also proves to be very helpful for our task as its morphological tag is all the information that is needed regarding the phrase.

A filtering stage follows which frees the corpus from noise like abbreviations, certain punctuation marks and other constituents that do not contribute to the SF detection task. Noun phrases inherit their part-of-speech tag (noun/adjective basically) from the part-of-speech value of their headword. Prepositional phrases and secondary clauses are labeled according to the preposition/conjunction introducing them respectively.

In our approach, the same verb may appear in the corpora in both the active and the passive voice. In this case the verb is considered to be two distinct verbs as its syntactic behavior may differ significantly depending on its voice.

5. Detection of verb environments

The environment of a verb is formed by the phrases preceding and following it. We name the number of these phrases the window size of the verb. We have carried out a number of experiments concerning the window size of a verb. Windows of sizes [-2+3], that is two phrases preceding and three phrases following the verb, [-2+2] and [-1+2] were tried. It is very likely for a correct frame to co-occur with one or more adjuncts in a real sentence and thus contain noise. As a consequence for almost every environment, not the entire environment (the entire window size), but a subset of the environment is a correct frame of the verb. Therefore all possible subsets (Sarkar and Zeman, 2000) of the above environments were produced by forming all possible permutations of their constituent-phrases. The frequency of a subset increases by adding to its count every time it is solidly formed in a permutation (see Figure 1).

In figure 2, we could distinguish two environments for the verb “αγγίζω” (to touch), environment A and B respectively. Environment A is consisted of a noun phrase in the nominative case (N1), a prepositional phrase (P5) and a noun phrase in the accusative case (N3), while B is consisted of two noun phrases, N1 and N2, in the nominative and the accusative case respectively. The parentheses next to each environment symbolize the number of occurrences in the corpus. For environment A, we calculate every single permutation of its subsets taking a left to right orientation, excluding duplicate ones (for example N1 P5 and not P5 N1). This is done in two steps, calculating the counts of the pair subsets first and then permutating them to obtain counts for the smallest subset available. The same procedure takes place for environment B. As a final stage, counts for every subset that exists in both environments are added (for example N1 occurred two times in environment A - after the permutations - and one time in environment B).

Upon completion of the procedure described above, we were able to formulate input data. We consider as potential SF both the original environment extracted from the corpora and all computed subsets as well.

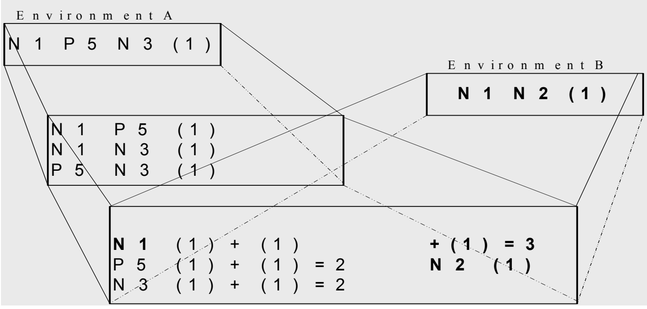

Figure 2. Subsets and their counts of environments A (in plain font) and B (in bold). By calculating every permutation of environments A and B respectively, we obtain the frequency of occurrence of every subset (shown in parentheses). As a last phase, frequencies of subsets of both environments are added.

5.1. Feature Selection

We intend to study the behavior of learning SF form large corpora by combining labeled and unlabeled examples. This section describes the parameters needed to formulate the vectors of data instances. Labeled and unlabeled data differ only in one variable, which is of course the class label.

Features of our training data were categorized in grammatical and numerical ones. Grammatical features consisted of the window size, varying from [-2+3] to [-1+2], along with seven more morphosyntactic categories, characterizing the type of phrase, the case, the preposition used, the presence or absence of an adverb, the type and tense of the verb.

As for the numerical features, our goal is to determine if a candidate SF is highly associated with a specific verb. To this end, the following counts are required:

- the count of a given environment with a given verb v, (k1).
- the count of a given verb v, (v1).
- the count of a given environment with every other verb except for v, (k2).
- the count of every other verb except for verb v, (v2).

Another numerical feature which plays a significant role is the Number of Distinct Elements (NDE) which is calculated as: NDE = number of distinct elements each environment has (for example: NDE(N1 P5 N3)=3).

5.2. Distinguishing the subject from the complements

Every valid frame may be a set of verb complements and optionally a subject (Unlike English, in Modern Greek the presence of the subject is not obligatory). After having identified the valid frames for a verb, the part of the frame that corresponds to the subject may be easily identified for non-copular verbs using empirical rules as it is always nominal and in the nominative case unlike the rest of the complements. For copular verbs, the predicate (also nominal and in the nominative case) normally can be also empirically distinguished from the subject by making use of information like its pos category (in most of the cases it is an adjective).

5.3. The LKB grammar development

LKB (Copestake and Flickinger, 2000) is a grammar development environment for grammars in a typed feature structure formalism that aims, however, to be independent of a particular linguistic framework. It has been implemented so that grammar loading time is short so as to enable easy validation of changes made to it. Having constructed a central type hierarchy for the grammar, the lexicon can be filled by verb entries containing subcategorization information we have acquired with the methodology described so far. Below follows the entry for the transitive verb ate for LKB. A set of square brackets denotes a feature structure.

```
ate : = verb &
[ SUBJ <[HEAD nominal &[CASE nom]]>,
COMPS  <[HEAD nominal &[CASE acc]]>].
```

6. Bayesian Networks

A Bayesian Belief Network (BBN) is a significant knowledge representation and reasoning tool, under conditions of uncertainty. (Mitchell, 1997). Given a set of variables D = <X1, X2…XN>, where each variable Xi could take values from a set Val(Xi), a BBN describes the probability distribution over this set of variables. We use capital letters as X,Y to denote variables and lower case letters as x,y to denote values taken by these variables. Formally, a BBN is an annotated directed acyclic graph (DAG) that encodes a joint probability distribution. We denote a network B as a pair B=<G,Θ>, (Pearl, 1988) where G is a DAG whose nodes symbolize the variables of D, and Θ refers to the set of parameters that quantifies the network. G embeds the following conditional independence assumption:

Each variable Xi is independent of its non-descendants given its parents.

Θ includes information about the probability distribution of a value xi of a variable Xi, given the values of its immediate predecessors. The unique joint probability distribution over <X_1, X_2…X_N> that a network B describes can be computed using:

$$P_B(X_1 ... X_N) = \prod_{i=1}^{N} P(x_i \mid parents\ (X_i))$$

6.1. Learning BBN from data

In the process of efficiently detecting verb SF, prior knowledge about the impact each feature has on the classification of a candidate SF as valid or not, is not straightforward. Thus, a BBN should be learned from the training data provided. Learning a BBN unifies two processes: learning the graphical structure and learning the parameters Θ for that structure. In order to seek out the optimal parameters for a given corpus of complete data, we directly use the empirical conditional frequencies extracted from the data (Cooper and Herskovits, 1992).

We use the following equation along with Bayes theorem to determine the relation *r* (or Bayes factor) of two candidate networks B_1 and B_2 respectively:

$$r = \frac{P(B_1 \mid D)}{P(B_2 \mid D)} \quad (1) \qquad P(B \mid D) = \frac{P(D \mid B) P(B)}{P(D)} \quad (2)$$

where:

P(B|D) is the probability of a network *B* given data *D*.

P(D|B) is the probability the network gives to data D.

P(D) is the 'general' probability of data.

P(B) is the probability of the network before seen the data.

Applying equation (1) to (2), we get:

$$r = \frac{P(D \mid B_1) P(B_1)}{P(D \mid B_2) P(B_2)} \quad (3)$$

Having not seen the data, no prior knowledge is obtainable and thus no straightforward method of computing *P(B$_1$)* and *P(B$_2$)* is feasible. A common way to deal with this is to assume that every network has the same probability with all the others.

The probability the model gives to the data can be extracted using the following formula (Glymour and Cooper, 1999):

$$P(D \mid B) = \prod_{i=1}^{n} \prod_{j=1}^{qi} \frac{\Gamma(\frac{\Xi}{q_i})}{\Gamma(\frac{\Xi}{q_i} + N_{ij})} \prod_{k=1}^{ri} \frac{\Gamma(\frac{\Xi}{r_i q_i} + N_{ijk})}{\Gamma(\frac{\Xi}{r_i q_i})}$$

where:

Γ is the gamma function.

n equals to the number of variables.

r_i denotes the number of values in *i:th* variable.

q_i denotes the number of possible different value combinations the parent variables can take.

N_{ij} depicts the number of rows in data that have *j:th* value combinations for parents of *i:th* variable.

N_{ijk} corresponds to the number of rows that have *k:th* value for the *i:th* variable and which also have *j:th* value combinations for parents of *i:th* variable.

Ξ is the equivalent sample size, a parameter that determines how readily we change our beliefs about the quantitative nature of dependencies when we see the data. In our study, we follow a simple choice inspired by Jeffreys (1939) prior. *Ξ* equals to the average number of values variables have, divided by 2.

Given the great number of possible networks produced by the learning process, a search algorithm has to be applied. We follow greedy search with one modification: instead of comparing all candidate networks, we consider investigating the set that resembles the current best model most.

In general, a BBN is capable of computing the probability distribution for any partial subset of variables, given the values or distributions of any subset of the remaining variables. Note that the values have to be discretised, and different discretisation size affects the network. As we shall discuss in the result section, BBN are a significant tool for knowledge representation, visualising the relationships between features and subsets of them. This fact has a significant result on identifying which features are actually affect the class variable, thus reducing training data size without any significant impact in the performance.

7. Support Vector Machines

Support Vector Machines (SVM) are in fact learning models designed to automatically trade-off accuracy and complexity by trying to minimize an upper bound on the generalization error (Cristianini *et al.*, 1998). The most important advantage of SVM is that contrary to other machine learning techniques, it behaves robustly even in high dimensional feature problems. SVM are based on the *Structural Risk Minimization* principle (Vapnik, 1995). The basic idea of this theory is to find a hypothesis *h* for which we could guarantee the lowest true error. By "true error" we denote the probability that a hypothesis will make an error when classifying a random unseen test vector. SVM are a new machine learning technique that have been applied to numerous classification and pattern recognition problems such as text classification, shallow parsing and face recognition with noteworthy results. (Joachims, 1996)

We focus on two-class classification problems. Let us denote a training set D as a pair $\{(x_i,y_i)\}$, i=1 to N with each input vector $x_i \in \Re^m$ and each binary label vector $y_i \in \{-1,+1\}$ corresponding to the two classes. SVM performs a mapping *φ* from the input space $\Re^m$ to the "feature" space $\Re^n$. In the case where data are linearly separable in $\Re^n$, a vector $w \in \Re^n$ ban be defined such that

$$w^T \phi(x) + b \geq 1 \qquad \text{if } y_i = 1 \qquad (4)$$

$$w^T \phi(x) + b \leq -1 \qquad \text{if yi} = -1 \qquad (5)$$

where $b \in \Re$ is a scalar.

A hyperplane $w^T \phi(x) + b$ is assembled for which the distance between itself and the positive and negative examples is maximized. We should also take into consideration that a hyperplane in space ℜn may represent a nonlinear decision surface in space ℜm. It can be shown (Cortes and Vapnik, 1996) that the vector w which will produce the "optimal" hyperplane, can be computed by minimizing $||w||^2$ and the resultant equation could be written as a linear combination of φ(x)'s . Thus, we obtain the following mathematical formulation:

$$w = \sum_{\iota=1}^{N} a_i y_i \varphi(\chi_\iota), \text{ where } a_i >= 0$$

Let us symbolize the vector of ai's as $A=(a_1 \ldots a_i \ldots a_N)$. A can be found by solving the following quadratic programming (QP) problem:

$$\text{Maximize } W(A) = A^T 1 - 1/2\, A^T QA \qquad (6)$$

Subject to: $A \geq 0$, $A^T Y = 0$ where $Y=(y_1 \ldots.. y_n)$ is a symmetric matrix with elements $Q_{ij} = y_i y_j \varphi(x_i)^T \varphi(x_j)$

In order to obtain Q_{ij}, we can find a kernel K(.,.) such that $K(x_i,x_j)=\varphi(x_i)T\varphi(x_j)$. In that way Q_{ij} becomes $y_i y_j K(x_i,x_j)$. As an example, note that the kernel of a polynomial classifier is $k(x_i,y_j)=(x_i^T x_j +1)^d$. Besides, notice that there are no local optima in (6) since Q is always positive and semi-definite. There are numerous kernel functions with good generalization capabilities.

For those examples from the training data along the margins of the decision boundary the corresponding a_i's are greater than zero (taken from Kuhn-Tucker theorem), these examples are called *support vectors*.

Concerning the testing process, provided a test vector $x \in \Re^n$, we first compute:

$$h = w^T \varphi(x) + b = \sum_{\iota=1}^{N} a_i y_i K(x \quad, x_j) + b$$

The class label for each x_i is assigned by applying the following empirical rule:

Label=1, if s>=s_0
Label=-1,if otherwise

The threshold s_0 is of course user defined. The SVM algorithm tries to minimize:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i$$

where parameters ξ_i measure the degree of margin constraints violation. They should satisfy equations (4), (5) when added to the right part of each. The penalty coefficient C is a user-defined parameter that controls the amount of '*slack'* allowed.

8. Significance of Unlabeled data

In the learning process, taking into consideration unlabeled data alone will result in a random, insufficient classifier, since there is no information about the class label (Castelli & Cover, 1995). Despite this fact, notice that unlabeled data do embed information about the joint distribution over features other than the class label (Nigam et al., 2000). This observation allows the use of unlabeled data along with a small set of labeled data to improve performance in certain domains.

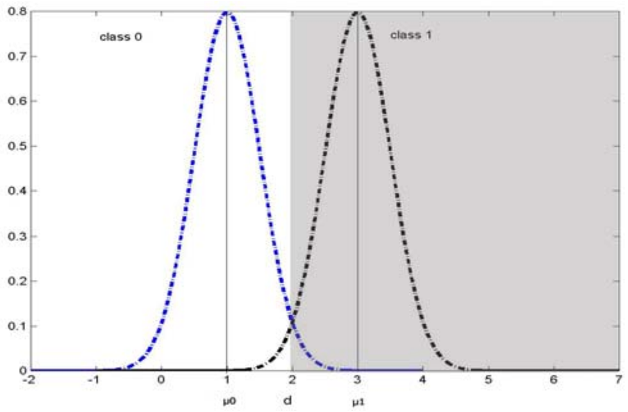

Figure 3. Data generated by two Gaussian distributions. Parameters of each distribution can be recovered using unlimited amount of unlabeled data. Class differentiation could then be learned using labeled data

To illustrate this, consider the following example. Suppose that we have instances that have been generated by a Gaussian mixture model. Figure 3 depicts that data are generated by two Gaussian distributions, one per class, with unknown parameters. The two classes are defined by the shaded and unshaded areas of the graph. The boundary d, which is called Bayes-optimal decision boundary and can classify instances into the two classes, could be calculated provided that one knows the Gaussian mixture parameters such as the mean and variance of each Gaussian distribution along with the mixing parameter between them. This example is analytically discussed by Nigam et al. (2000).

Suppose now that we have an unlimited number of unlabeled data while only a finite number of labeled ones. It has been proven (McLachlan & Krishnan 1997, section 2,7) that the Gaussian distribution parameters could be retrieved from unlabeled data alone, provided that they have been originated from a Gaussian mixture model. Nevertheless, without labeled data it is impossible to assign the class labels to each of the distributions. Castelli & Cover, (1995) argued that the problem of using labeled data to determine the class of a Gaussian distribution converges exponentially quickly in the number of labeled training instances.

It is noteworthy to point out that the above statements rely on the critical assumption that both labeled and unlabeled data have been originated by the same parametric model. However, this restriction rarely holds in real-world domains, such as the identification of verb subcategorization frames from large corpora, where semantics of the language play an important role in the decision of whether a candidate SF is valid or not. This argument raises the issue of using unlabeled data in cases where assumptions are being violated

For our task, we exploit the ability of Bayesian networks to encode prior knowledge and the classification ability of SVM. Applying BBN to SVM is quite straightforward. First, we only use the labeled data set, denoted by S_l, along with the prior knowledge mentioned above to learn the most probable network B_l. We then try to classify the set of unlabeled instances S_{un} using the conditional probability table associated with B_l. The new classified set $S`_{un}$ is merged with S_l to form $\mathrm{S} = \mathrm{S}_1 + \mathrm{S}'_{\mathrm{un}}$. The objective now is to learn a new network B_S from the unified data set S given again the same prior knowledge as before. This could be done without any risk since this kind of knowledge is uniform to all instances. Upon completion of this procedure, we are able to estimate the prior distribution of *θ(x)*. This distribution is used to define the parameters of the kernel that we will use. For our approach, we used the polynomial kernel. As a final step, we train a SVM classifier from *S*, using the calculated prior distribution *θ(x)*

9. Experimental Results

The corpora used for our study are the balanced ILSP/ELEFTHEROTYPIA Greek Corpus (consisting of 1.6 million words of political, social and sports content, taken from a wide circulation newspaper), the balanced ESPRIT-860 Corpus for Modern Greek of approximately 300.000 words and a significant part of the corpus of economic news created for the DELOS project consisting of approximately 32M of raw text (Sintichakis et al., 2000).

We define precision as the percentage of correct frames to all the frames which were acquired.

$$precision = \frac{\#\,of\,correctly\,identified\,frames}{\#\,of\,positive\,predictions}$$

Table 2 tabulates the experimental results obtained from both corpora using Bayesian Networks, Support Vector Machines and our hybrid algorithm of using Bayesian inference knowledge and inserting it into the SVM hyperparameters, noted as B-SVM. For the task of identifying a valid SF using additional unlabeled examples, we have conducted the experiments using approximately 6.500 unlabeled instances from each corpus. Additionally, in order to obtain a more inclusive view of the task, we provide results using statistical machine learning algorithms such as relative frequency

(RF), T-score and LLR metrics (Dunning, 1993).

DELOS	METHOD					
Window	**BBN**	**SVM**	**B-SVM**	**RF**	**T**	**LLR**
[-1+2]	70.4	73.3	75.8	43.1	57.5	68.1
[-2+2]	72	75.3	78.9	43.6	59	69.3
[-2+3]	71.3	74.6	78.3	43.3	58	68.5
ILSP	**METHOD**					
Window	**BBN**	**SVM**	**B-SVM**	**RF**	**T**	**LLR**
[-1+2]	83.9	87.2	90	79.4	71	78.2
[-2+2]	86.3	90	94.2	82.8	73	80.1
[-2+3]	85.7	89	93.4	80.3	72.6	78.7

Table 2: Experimental results obtained from the two corpora

By observing the obtained results, we could claim that both BBN and SVM perform significantly better than the other machine learning algorithms by a factor that varies from 5 to almost 30%, a fact that supports the argue that BBN and SVM are well suited for the task of verb subcategorization identification (Maragoudakis et al, 2001). Furthermore, by incorporating bayesian knowledge into the SVM classifier and using a set of unlabeled examples, we achieve a 3-6% improvement.

The size of the window is also found to be of great importance when dealing with verb subcategorization. As demonstrated by the results, a window size of [-2+2] is the best choice for MG SF detection. Another interesting observation is that DELOS corpus performs worse than ILSP. There are two possible reasonable explanations; DELOS corpus is an economic corpus with a morphologically narrower set of elements surrounding a verb. The other reason is that economists use to develop their own terms and expressions, thus making it difficult for an automatic system to adjust, unless these terms co-occur with the verbs many times.

10. Conclusion

The identification of verb SF could contribute to the significant improvement of natural language human computer interaction systems since they could embed important information about the syntactic-semantic constituents of a verb. This process is considered to be a shallow parsing task due to the fact that only specific parts of the syntactic information of a verb are extracted. This paper has presented a group of machine learning algorithms that aim to address the issue of automatically learning verb subcategorization frames from large text corpora.

New frames not known beforehand were learned throughout the training process. More specifically, Bayesian belief networks, Support Vector Machines, Relative Frequencies, T-score and Log Likelihood Ratio were applied to the task at hand, using balanced as well as domain-specific corpora. Using minimal linguistic resources, i.e. basic morphological tagging and phrase chunking, verb environments were identified and every environment subset was formulated. Thus, the need for fully annotated input data is alleviated.

Given the obvious high cost of hand-labeling the data and the vast volumes of available text (in the web, in corpora, etc.), a novel method of using unlabeled examples to supplement limited labeled data has been studied. Parameters needed by SVM are learned via BBN learning from labeled and unlabeled examples. The resulting classifier outperforms all other techniques mentioned. Our idea can be used for other languages as well with slight modifications. For example, for a not "free word order" language one would only have to calculate permutations of an environment treating duplicate pairs as different (N1 P5≠P5 N1).

We have shown how the information obtained about the arguments of a verb can be used to fill the corresponding fields of an HPSG lexicon in the LKB grammar development environment. A next step would be the automatic detection of verb selectional restrictions so as to provide a more complete picture concerning verb local dependencies.

11. References

Androutsopoulos, I. and R. Dale. 2000. Selectional Restrictions in HPSG. Proceedings of the 18th International Conference on Computational Linguistics, COLING 2000, Saarbruecken, Germany, 1:15-20.

Basili R., Pazienza M.T. and Vindigni M. 1997. Corpus-driven Unsupervised Learning of Verb Subcategorization frames. Proceedings of the Conference of the Italian Association for Artificial Intelligence, AI*IA 97, Rome.

Brent M. 1993. From Grammar to Lexicon: Unsupervised Learning of Lexical Syntax. Computational Linguistics, vol. 19, no. 3, pp. 243-262.

Briscoe T. and Carroll J. 1997. Automatic Extraction of Subcategorization from Corpora. Proceedings of the 5th ANLP Conference, pp.356-363. ACL, Washington D.C.

Castelli V. and Cover T. 1995 On the exponential value of labeled samples. Pattern Recognition Letters, 16(1), 105-111.

Cooper J. and Herskovits E. 1992. A Bayesian method for the induction of probabilistic networks from data. Machine Learning, 9, pp.309-347.

Cortes C. and Vapnik V. 1996. Support vector networks. Machine Learning, 20,pp 273-297

Cristianini N Shawe-Taylor J. and Campbell C. 1998. Dynamically adapting kernels in support vector machines. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, Advances in Neural Information Processing Systems, 11. MIT Press, 1998

De Lima F. 1997. Acquiring German Prepositional Subcategorization frames from Corpora. Proceedings of the 5thWorkshop on Very Large Corpora (WVLC-5).

Dunning T. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. Computational Linguistics. vol.19, no. 1, pp. 61- 74.

Eckle J. and Heid U. 1996. Extracting raw material for a German subcategorization lexicon from newspaper text. Proceedings of the 4th International Conference on Computational Lexicography, COMPLEX'96, Budapest, Hungary.

Gahl S. 1998. Automatic extraction of subcorpora based on subcategorization frames from a part-of-speech tagged corpus. Proceedings of COLING-ACL 1998, pp.428-432.

Gao J.B., Gunn S.R., Harris C.J. and Brown M. 2001. A probabilistic framework for SVM regression and error bar estimation. Machine Learning, Submission

Glymour C. and Cooper G. (eds.). 1999. Computation, Causation & Discovery. AAAI Press/The MIT Press, Menlo Park.

Jeffreys H. 1939. Theory of Probability. Clarendon Press, Oxford.

Joachims T 1996. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. Carnegie Mellon University, Technical Report, pp 96-118

Kawahara D., Kaji N. and Kurohashi S. 2000. Japanese Case Structure Analysis by Unsupervised Construction of a Case Frame Dictionary. Proceedings of COLING 2000.

Kohavi R. 1995. The power of Decision Tables. The European Conference on Machine Learning (ECML)

Kohonen T. 1987. Self-Organization and Associative Memory. 2nd Edition, Berlin: Springer-Verlag.

Korhonen A., Gorrell G. and McCarthy D. 2000. Statistical Filtering and Subcategorization Frame Acquisition. Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. Hong Kong.

Manning C. 1993. Automatic Acquisition of a Large Subcategorization Dictionary from Corpora. Proceedings of 31st Meeting of the ACL 1993, pp. 235-242.

Maragoudakis M., Kermanidis K., Fakotakis N. and Kokkinakis G. 2000. Learning Automatic Acquisition of Subcategorization Frames using Bayesian Inference and Support Vector Machines. First International Conference on Data Mining, San Jose, to appear.

McLachlan G., Krishman T., 1997. The EM Algorithm and Extensions. John Wiley and Sons, New York.

Mitchell T. 1997. Machine Learning. Mc Graw-Hill

Nigam K., McCallum A., Thrun S. and Mitchell T. Text Classification from Labeled and Unlabeled Documents using EM. Machine Learning, 39(2/3). pp. 103-134. 2000

Osuna E., Freud R. and Girosi F.1997. Training support vector machines: an application to face detection. Proceedings of computer vision and pattern recognition .Puerto Rico

Pearl J. 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Mateo, CA: Morgan Kaufmann.

Sag, I.A. and T. Wasow. 1999. Syntactic Theory. A Formal Introduction. CSLI Publications, Stanford California.

Sarkar A. and D. Zeman, 2000. Automatic Extraction of Subcategorization Frames for Czech. In Proceedings of the 18th International Conference on Computational Linguistics, pp.691-697.

Sgarbas K., Fakotakis N. and Kokkinakis G. 1999. A Morphological Description of Modern Greek using the Two-Level Model (in Greek). Proceedings of the 19th Annual Workshop, Division of Linguistics, University of Thessaloniki, Greece, April 23-25, 1999, pp.419-433.

Sintichakis M., Kermanidis K., Kalamboukis T. 2000. Corpus Analysis for Applied Lexicography. Proceedings of COMLEX 2000, pp. 121-126, Kato Achaia, Greece.

Sollich P. 2001. Bayesian methods for Support Vector Machines: Evidence and predictive class probabilities. Machine Learning, to appear

Stamatatos E., Fakotakis N. and Kokkinakis G. 2000. A Practical Chunker for Unrestricted Text. Proceedings of the 2nd International Conference of Natural Language Processing (NLP2000), pp. 139-150.

Vapnik V. 1995 The Nature of Statistical Learning Theory. Springer, New York