A Bayesian Model for Shallow Syntactic Parsing of Natural Language Texts

Manolis Maragoudakis, Nikos Fakotakis and George Kokkinakis
Intelligent Systems Group
University of Patras
Rion 26500, Patras, Greece
{mmarag,fakotaki,gkokkin}@wcl.ee.upatras.gr

Abstract

For the present work, we introduce and evaluate a novel Bayesian syntactic shallow parser that is able to perform robust detection of pairs of subject-object and subject-direct object-indirect object for a given verb, in a natural language sentence. The shallow parser infers on the correct subject-object pairs based on knowledge provided by Bayesian network learning from annotated text corpora. The DELOS corpus, a collection of economic domain texts that has been automatically annotated using various morphological and syntactic tools was used as training material. Our shallow parser makes use of limited linguistic input. More specifically, we consider only part of speech tagging, the voice and the mood of the verb as well as the head word of a noun phrase. For the task of detecting the head word of a phrase we used a sentence boundary detector. Identifying the head word 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. The evaluation of the proposed method was performed against three other machine learning techniques, namely naïve Bayes, k-Nearest Neighbor and Support Vector Machines, methods that have been previously applied to natural language processing tasks with satisfactory results. The experimental outcomes portray a satisfactory performance of our proposed shallow parser, which reaches almost 92 per cent in terms of precision.

1. Introduction

Throughout the recent years, there has been an increasing interest in corpus-based natural language processing using machine learning approaches. Numerous algorithms have been applied to large corpora, aiming at extracting the essential syntactic or semantic information rather than deriving a detailed syntactic-semantic analysis of each sentence. As manual construction of resources containing linguistic information is laborious and time consuming, the recent trend is the mining of such information automatically from textual corpus data. This task of identifying and extracting the key parts of information from a sentence is called *shallow parsing*. Characteristic examples of shallow parsing tasks include the identification of noun phrases, text chunking that recognizes the type of a phrase, the extraction of subject, main verb and object, etc. The majority of previous approaches to syntactic information acquisition have made use of sophisticated linguistic resources and pre-processing tools (syntactic treebanks, wide coverage parsers etc.). As, for the majority of languages (including Modern Greek), such resources are not yet available, acquiring the necessary information by deploying as limited linguistic resources as possible appears to be very challenging. For the present work, preprocessing of the input corpus reaches merely the stage of elementary intrasentential, non-embedded phrase chunking.

The present work introduces a novel, Bayesian shallow parser that learns pairs of subject-verb-object and subject-verb-direct object-indirect object from large corpora. This work was funded under the DELOS project (EPETII-98LE-24), which deals with the construction of a lexicon of economic terminology for Modern Greek (MG). We applied the proposed method to large economic corpora (approximately a hundred Megabytes of raw text) in order to enrich the syntactic information of a verb's lexical entry. Apart from the lexical value of the syntactic information, the shallow parser outcome could also provide consequential information to dialogue systems, since they determine the agent and the receiver of an action. Furthermore, shallow parsing is essential for dialogue systems since it does not require high computational effort, thus it allows for making dialogue flow quicker. Current spoken dialogue systems employ simple linguistic processing techniques due to the real time performance constraint. Therefore, we conclude that our approach provides a tool for the instant detection of subject-verb-object, which is considered essential to the efficiency of the dialogue.

The structure of the paper is as follows: Section 2 introduces the linguistic resources used in our task, while Section 3 provides a discussion on Bayesian networks. Section 4 presents the implementation issues and Section 5 concludes with the experimental results.

2. Corpus and Linguistic Preprocessing

The DELOS Corpus (Kermanidis, Fakotakis and Kokkinakis 2002) is a collection of economic domain texts of approximately five million words and of varying genre (press reportage, news, articles, interviews and scientific studies). It has been automatically annotated from the ground up: morphological tagging on DELOS was performed by an analyzer for Modern Greek based on Koskenniemi's two-level morphology model and utilizes a lexicon of more than 60,000 lemmata (Sgarbas, Fakotakis and Kokkinakis 2000). The provided information includes Part-Of-Speech (POS) tagging for all words, case tagging for nouns, adjectives and pronouns, voice tagging for verbs, type tagging for verbs (distinguishing between personal and impersonal verb types), type tagging for pronouns (distinguishing among relative, interrogative and the rest of the pronouns) and type tagging for conjunctions (distinguishing between coordinating and subordinating conjunctions). Precision and recall values in pos tagging reach 84-88% and 95-98% respectively. Concerning some key morphological features, case tagging reaches an accuracy exceeding 94%, and voice

tagging for verbs 84%. Further (phrase structure) information is obtained automatically by the chunker described in detail in Stamatatos, Fakotakis and Kokkinakis (2000). Noun (NP), verb (VP), prepositional (PP), adverbial phrases (ADP) and conjunctions (CON) are detected via multi-pass parsing. The chunker is based on a small keyword lexicon containing some 450 keywords (articles, pronouns, etc.) and a suffix lexicon of 300 of the most common word suffixes in Modern Greek. Smaller phrases (simple NPs, PPs and VPs) are formed in the first passes, while later passes combine smaller phrases to form more complex structures (ADPs, CONs, coordinate structures, attachment of genitives to the preceding phrase). As reported in Stamatatos et al. (2000), the precision of the chunker reaches 94.5% and recall 89.5% when tested on a corpus of 200,000 words of the Modern Greek newspaper *TO BHMA* (The Tribune). The phrase headword is identified next. Noun phrase headwords are detected based on a set of empirical rules: scanning through the constituents, the head-word is, in the following priority, the noun, adjective or numeral. Regarding the case, the priority is: nominative, accusative, genitive. In case the noun phrase does not contain any of the above pos categories, the head-word is any word starting with a capital letter. For verb phrases, the headword is the main verb, unless they are introduced by a conjunction in which case the conjunction is the headword. For prepositional phrases it is the preposition introducing them and for adverbial phrases it is the first word of the phrase.

3. Bayesian Networks

Classification is a fundamental concept in the fields of data mining and pattern recognition that requires the construction of a function that assigns a target or class label to a given example, described by a set of attributes. This function is referred to as a *classifier*. Given a set of pre-classified instances, numerous machine learning algorithms such as neural networks, decision trees, rules and graphical models, attempt to induce a classifier, able to generalize over the training data.

While Bayesian graphical models were known for being a powerful mechanism for knowledge representation and reasoning under conditions of uncertainty, is was only after the introduction of the so-called naïve Bayesian classifier (Duda and Hart, 1973; Langley, Iba and Thomson, 1992) that they were regarded as classifiers, with a prediction performance similar to state-of-the-art classifiers. The naïve Bayesian classifier performs inference by applying Bayes rule to compute the posterior probability of a class C, given a particular vector of input variables A_i. It then outputs the class whose posterior probability is the highest. Regarding its computational cost, inference in naïve Bayes is feasible, due to two assumptions, yet often unrealistic for real world applications:

- All the attributes A_i are conditionally independent of each other, given the classification variable.
- All other attributes are directly dependent on the class variable.

Despite the fact that naïve Bayes performs well, it is obviously counterintuitive to ignore the correlation of the variables in some domains. As an example, consider the credit card fraud detection problem (Heckerman, 1995). It would be hard to assume that the age or the sex of a person who is suspect for being involved in a credit card fraud is uncorrelated with the purchase of gas or jewellery.

Bayesian networks (Pearl, 1988) provide a comprehensive means for effective representation of independence assumptions. They are capable of effectively coping with the non-realistic naïve Bayes restriction, since they allow stating conditional independence assumptions that apply to all or to subsets of the variables. A Bayesian network is consisted of a qualitative and quantitative portion, namely its structure and its conditional probability distributions respectively. Given a set of attributes $A=\{A_1,...,A_k\}$, where each variable A_i could take values from a finite set, a Bayesian network describes the probability distribution over this set of variables. We use capital letters as X,Y to denote variables and lower case as x,y, to denote values taken by these variables. Formally, a Bayesian network is an annotated directed acyclic graph (DAG) that encodes a joint probability distribution. We denote a network B as a pair $B=<S,P>$ (Pearl, 1988) where S is a DAG whose nodes correspond to the attributes of A. P refers to the set of probability distributions that quantifies the network. S embeds the following conditional independence assumption:

Each variable A_i is independent of its non-descendants given its parent nodes.

P includes information about the probability distribution of a value a_i of variable A_i, given the values of its immediate predecessors in the graph, which are also called *parents*. This probability distribution is stored in a table, which is called conditional probability table. The unique joint probability distribution over A that a network B describes, can be computed using:

$$p_B(A_1,...,A_n) = \prod_{i=1}^{n} p(A_i \mid parents(A_i)) .$$

The network of figure 1 encodes a probability distribution which is estimated as follows:

$p(A_1,A_2,A_3,A_4,A_5)=p(A_2|A_1)p(A_3|A_1,A_4)p(A_5|A_3)$

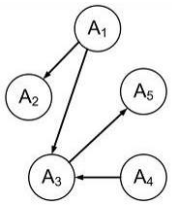

Figure 1: A general, unrestricted Bayesian network

A noteworthy remark is that the naïve Bayes classifier is actually a simple Bayesian network with a fixed, unique structure. The class node is a parent to all attribute nodes and there are no arcs between the attribute nodes (figure 2). This structure captures the two assumptions of naïve Bayes.

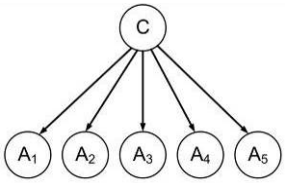

Figure 2: Naïve Bayesian classifier as a Bayesian network

3.1 Learning Bayesian Networks from Data

There are two practices for determining the structure of a Bayesian network. Either manually, by a human domain expert who should provide the interconnection of the variables, or having the structure determined automatically by learning from a set of training examples. Regarding the learning of the conditional probability table of a network, the same principle applies. The parameters of the table could either be provided manually by an expert or automatically through a learning procedure. The task of manually supplying the parameters is a laborious one. Besides, in some applications it is simply infeasible for a human expert to know a priori both the structure and the conditional probability distributions. The problem of finding the most probable network structure from data is known to be NP-hard (Mitchell, 1997). The most commonly utilized approach is the introduction of a scoring metric that evaluates the probability of a candidate structure *B* over the training set *D*. The two standard metrics used to learn networks from data are the *Bayesian scoring function* (Cooper and Herskovits, 1992) and the one which is based on the principle of *minimal description length* (MDL) (Suzuki, 1993; Lam and Bacchus, 1994; Friedman, Geiger and Goldszmidt, 1997). Nevertheless, Heckerman (1995) observed that the two metrics are asymptotically equivalent as the sample size increases. Furthermore, they prove to be asymptotically correct, meaning that with probability one, the learned distribution converges to the underlying distribution as the number of training instances increases.

4. Implementation

As regards training data, we used approximately 10000 sentences in which the tuples of subject-verb-object and subject-verb-direct object-indirect object were manually annotated by experienced linguists. The morphological information was extracted from the DELOS corpus provided that the linguistic tools, described in section 2, were applied. Finally, the chunker was incorporated in order to mine the phrase boundaries and the head word of noun and verb phrases. From the plethora of available linguistic information that the DELOS corpus contains, the Bayesian shallow parser exploits only part of speech tagging, the voice-mood of the verb, the case of the head word of a noun phrase as well as the phrase boundaries.

Upon completion of the annotation procedure, we performed Bayesian network learning from data using the *Bayesian scoring function*. Since the population of candidate networks that may reflect the probability distribution of data becomes cumbersome, a search algorithm had to be followed: Initially, the most probable forest-structured network is constructed (i.e. a network in which every node has at most one parent). A greedy search is performed by adding, deleting or reversing the arcs randomly. In case that a change results in a more probable network, that network is accepted, otherwise cancelled. Throughout this process, a repository of networks with high probability is maintained. When the search reaches a local maximum, a network is randomly selected from the repository and the search process is activated again. It should be noted that in order to avoid the convergence to the previous local maximum the network is slightly modified, meaning that we delete some arcs. Since the training data set is large we also sub-sample the data to speed the network evaluation process up. During the search, the size of the sub-samples is increased. The network complexity is also controlled during the search, so that a limited number of arcs is allowed in the beginning and as the process progresses, more and more arcs are approved. Recall that given two nodes X and Y of x and y discrete states each, the conditional probability table of a network $X \rightarrow Y$ will store at least $x \cdot y$ parameters. It is important to penalize huge tables, corresponding to fully-connected networks, which is the most naïve way of learning. These two annealing schemes (sub-sampling and complexity restrictions) have proven to have the effect of avoiding many bad local maxima (Heckerman, Geiger and Chickering, 1995). The extracted, most probable network is then used in order to estimate whether a candidate noun phrase is a subject or an object (direct or indirect) for a given verb.

5. Evaluation

Concerning the evaluation process, we have developed prototype simulation software to be used by linguists in order to establish the unbiased behavior of the Bayesian shallow parser in real world applications such as the DELOS project. We have applied the shallow parser to approximately 20000 sentences and arbitrary selected about 5000 of them for manual evaluation. Experimental results are tabulated in table 1. The limited accuracy of the preprocessing tools (e.g. POS tagger, morphological analyzer) brings about an error in the performance of our method. Therefore, we provide detailed error analysis of each unit along with its impact factor in the performance of the proposed methodology.

Error type	# of sentences	Percentage
Correct	*3648*	*73%*
Errors in case	*288*	*5.7%*
Errors in POS	*378*	*7.5%*
Errors in the number of the head word	*160*	*3.2%*
Errors of head word	*121*	*2.5%*
Errors of Bayesian shallow parser	*405*	*8.1%*
Total	**5000**	**100%**

Table1: Overview of the Bayesian shallow parser evaluation outcome

A close look at table 1 reveals that the error rate appears to be approximately 27%. However, almost the 70% of those errors (947 cases out of a total of 1352 erroneous ones) is caused due to erroneous output of the pre-processing modules, as regards to the tagging of the head word of a noun phrase which is detected (or actually is) as subject or object. The most significant influence appears to originate from errors referring to the POS, the case, the number of the head word or even the identification of the head word. Provided error-free output from the other linguistic tools, the shallow syntactic parser appears to perform robustly.

Regarding the evaluation of our Bayesian module in contrast to other machine learning algorithms, we applied a 10-fold cross validation (Stone, 1974) using three other well known methods that have been previously applied to various computational linguistics tasks, demonstrating significant results (Maragoudakis et al., 2001). More respectively, naïve Bayes, k-Nearest Neighbor (k-NN) and Support Vector Machines (SVM) have been used in addition to our Bayesian shallow parser. The three above mentioned methods derive from different theoretical backgrounds, varying from probabilistic theory to instance based learning and structural risk minimization.

Three indexes of performance that are commonly used in supervised learning tasks have been used, namely precision, recall and F-measure. Precision (p) is defined as the number of correctly identified pairs of subject-object of a verb (tp), divided by the number of correctly identified pairs, plus the number of incorrectly selected cases (fp) for that verb:

$$p = \frac{tp}{tp + fp}$$

Recall (r) is estimated as the number of correctly identified pairs of subject-object of a verb (tp), divided by the number of correctly identified pairs plus the number of cases the system failed to classify for that verb (fn):

$$r = \frac{tp}{tp + fn}$$

The F-measure (f) is the harmonic mean of precision and recall, calculated as:

$$f = 1 \Big/ \left(a\frac{1}{p} + (1-a)\frac{1}{r} \right),$$

where α is a factor which determines the equilibrium of precision and recall. A value of α=0.5 is often chosen for equal weighting of precision and recall. Figure 3 outlines the outcomes of the above mentioned methods.

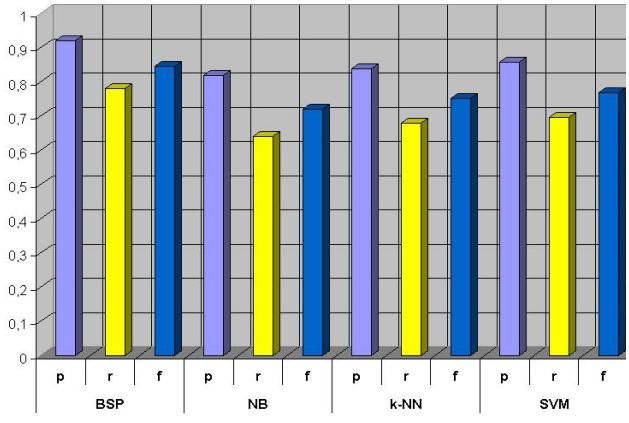

Figure 3: Precision, Recall and F-measure for all the applied machine learning algorithms, using 10-fold cross validation.

Note that, BSP refers to the Bayesian shallow parser, NB stands for naïve Bayes, while k-NN and SVM correspond to the k-Nearest Neighbor and Support Vector Machines respectively. As one may observe, BSP outperforms all other methods both in precision and recall, a fact that supports our claim that Bayesian networks theory is well suited for such applications. More respectively, the precision of BSP reaches 92% and the recall metric is close to 79%. SVM are slightly inferior than BBN by a factor of 9%. The k-NN algorithm is 12% worse, while the NB appears to be the most error-prone classifier. This may be attributed to the fact that the naïve Bayesian classifier is based on clearly over-restrictive assumptions.

Bibliographical References

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

Duda, R. and Hart, P. (1973). Pattern Classification and Scene Analysis. New York: John Wiley & Sons.

Friedman, N., Geiger, D. & Goldszmidt, M. (1997). Bayesian network classifiers. Machine Learning.

Heckerman, D. (1995). A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Redmond, Washington.

Heckerman, D., Geiger, D. & Chickering, D. (1995). Learning Bayesian Networks: the Combination of Knowledge and Statistical Data, Machine Learning 20, 197-243.

Kermanidis, K., Fakotakis, N. & Kokkinakis, G. (2002). DELOS: An automatically tagged economic corpus for Modern Greek. Proceedings of LREC 2002, (pp. 93-100). Las Palmas de Gran Canaria.

Sgarbas, K., Fakotakis, N. & Kokkinakis, G. (2000). A straightforward approach to morphological analysis and synthesis. Proceedings of the COMLEX 2000 Workshop, (pp. 31-34), Kato Achaia, Greece.

Lam, W. & Bacchus, F. (1994). Using New Data to Refine a Bayesian Network, In Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, (pp. 383-390) Morgan Kaufmann Publishers, San Mateo, California.

Langley, P., Iba, W. & Thomson, K. (1992). An analysis of Bayesian classifiers, in Proc. 10th Ntl Conf in AI, (pp. 223-228).

Maragoudakis, M., Kermanidis, K., Fakotakis, N. & Kokkinakis, G. (2001). Learning Automatic Acquisition of Subcategorization Frames using Bayesian Inference and Support Vector Machines. The 2001 IEEE International Conference on Data Mining (pp 301-304), November 29 - December 2, 2001, San Jose.

Mitchell, T. (1997). Machine Learning, Mc Graw-Hill.

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

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

Stone, M. (1974). Cross-validation choice and assessment of statistical predictions. Journal of the Royal Statistical Society, (36), (pp.111-147).

Suzuki, J. (1993). A construction of Bayesian networks from databases on a MDL scheme. In Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence, (pp. 266-273), San Francisco.