Decision Trees and NLP: A Case Study in POS Tagging

Giorgos Orphanos, Dimitris Kalles, Thanasis Papagelis and Dimitris Christodoulakis

Computer Engineering & Informatics Department
and Computer Technology Institute
University of Patras
26500 Rion, Patras, Greece
{georfan, kalles, papagel, dxri}@cti.gr

ABSTRACT

This paper presents a machine learning approach to the problems of part-of-speech disambiguation and unknown word guessing, as they appear in Modern Greek. Both problems are cast as classification tasks carried out by decision trees. The data model acquired is capable of capturing the idiosyncratic behavior of underlying linguistic phenomena. Decision trees are induced with three algorithms; the first two produce generalized trees, while the third produces binary trees. To meet the requirements of the linguistic datasets, all three algorithms are able to handle set-valued attributes. Evaluation results reveal a subtle differentiation in the performance of the three algorithms, which achieve an accuracy range of 93-95% in POS disambiguation and 82-88% in guessing the POS of unknown words.

INTRODUCTION

It has recently become apparent that empirical ML can find in NLP an exciting application area. The increasing use of corpus-based learning in place of manual encoding has led to the rebirth of empiricism in NLP, with primary goal to overcome a perennial problem, namely the *linguistic knowledge acquisition bottleneck*: for each new, different or slightly different task of NLP, linguistic knowledge bases (lexicons, rules, grammars) most of the time have to be built from scratch. An additional reason to pursue automatically acquired language models is that it is practically impossible to manually encode all the exceptions or sub-regularities occurring even in simple language problems, or give emphasis to the most frequent regularities.

Corpus-based approaches have been successful in many areas of NLP, but it is often the case that language is being treated like a black-box system simulated by large tables of statistics. Although, from the engineering point-of-view, it is a wide-spread practice to consider systems as black boxes, it is obvious that this opaqueness makes it difficult to understand and analyze underlying linguistic phenomena and, consequently, the improvement of the language model may depend on parameters irrelevant to the language itself. This disadvantage has been the main source of criticism against the purely statistical approaches.

The optimism about the marriage of ML and NLP stems from the observation that most NLP problems can be viewed as *classification* problems (Magerman, 1995; Daelemans, 1997). Empirical learning is fundamentally a classification paradigm and, as stated in (Daelemans, 1997), the point is to redefine linguistic tasks as classification tasks. In general, linguistic problems fall into two types of classification: (a) Disambiguation, i.e., determine the correct category from a set of possible categories and (b) Segmentation, i.e., determine the correct boundary of a segment from a set of possible boundaries. Some examples of disambiguation are: (i) determine the pronunciation of a letter, given its neighboring letters, (ii) determine the part-of-speech (POS) of a word with POS ambiguity, given its contextual words, (iii) determine where to attach a prepositional phrase, given a set of other phrases, (iv) determine the contextually appropriate meaning of a polysemous word. Some examples of segmentation are: (i) given a letter in a word, determine whether the word can be hyphenated after that letter, (ii) determine if a period is the boundary of two sentences, (iii) determine the boundaries of the constituent phrases in a sentence.

This paper focuses on the empirical learning of two NLP tasks performed by POS taggers, viz. *POS disambiguation* and *unknown word guessing*, both viewed as tasks of disambiguation. The target language is Modern Greek (M. Greek), a natural language which, from the computational perspective, has not been as widely investigated. In (Orphanos and Tsalidis, 1999) we have shown the successful application of automatically induced decision trees to the problems of POS disambiguation and unknown word guessing, as they appear in M. Greek. In this paper we describe three algorithms for decision tree induction and compare their performance on the above linguistic problems. The first two algorithms produce generalized decision trees, while the third produces binary decision trees and uses pre-pruning techniques to increase generalization accuracy. All three algorithms are able to handle **set-valued** attributes, a requirement posed by the nature of the linguistic datasets. Our experiments exhibit a performance range of 93-95% in POS disambiguation and 82-88% in guessing the POS of unknown words.

The structure of this paper is as follows: In the next section we give an overview of POS tagging techniques. Then, we present the decision tree approach applied to POS tagging, with emphasis to M. Greek, and describe three tree induction algorithms. Consequently, we give a detailed description of the datasets used for the training algorithms and illustrate detailed performance measurements. Finally, we discuss the performance of the decision-tree approach to POS disambiguation/guessing and compare the results achieved by the three algorithms.

OVERVIEW OF POS TAGGING TECHNIQUES

POS taggers are software devices that aim to assign unambiguous morphosyntactic tags to words of electronic texts. Their usefulness to the majority of natural language processing applications (e.g., syntactic parsing, grammar checking, machine translation, automatic summarization, information retrieval/extraction, corpus processing, etc.) has led to the evolution of various techniques for the development of robust POS taggers. Although the hardest part of the tagging process is accomplished by a computational lexicon, a POS tagger cannot solely consist of a lexicon due to: (i) morphosyntactic ambiguity (e.g., 'love' as verb or noun) and (ii) the existence of unknown words (e.g., proper nouns, place names, compounds, etc.). When the lexicon can assure high coverage, unknown word guessing can be viewed as a decision taken upon the POSs of open-class words.

The first corpus-based attempts for the automatic construction of POS taggers used hidden Markov models (HMMs), which were borrowed from the field of speech processing (Bahl and Mercer, 1976; Derouault and Marialdo, 1984; Church, 1988). HMM taggers, also known as *n-gram* taggers, make the drastic assumption that only the *n-1* words have any effect on the probabilities of the next word (a common *n* is 3, hence the term *trigrams*). While this assumption is clearly false, surprisingly n-gram taggers can obtain very high rates of tagging accuracy, ranging from about 95% to 98%. Due to their high accuracy, n-gram taggers have come to be standard and are available for many languages. Dermatas and Kokkinakis (1995) have trained n-gram taggers for seven European languages, viz. English, Dutch, German, French, Greek, Italian and Spanish.

Another approach utilizes neural networks for tagging, which, as reported in (Schmid, 1994a), can achieve equal or better accuracy compared to HMM approach (yet with lower processing speed). However, both approaches treat language as a black box filled with probabilities and transition weights. Other lines of development use methods that try to capture linguistic information directly and thus provide the ability to model underlying linguistic behavior with more comprehensive means. Under this concept one can find the linguistic (manual) approach, where experts encode handcrafted rules or constraints based on abstractions derived from language paradigms (Green and Rubin, 1971; Voutilainen 1995). The amount of effort required by the manual approach and its inherent inflexibility led to the pursuit of ML techniques for the automatic induction of disambiguation rules (Hindle, 1989; Brill, 1995), or equivalent inference devices such as decision trees (Schmid, 1994b; Daelemans *et al.*, 1996) or decision lists (Yarowsky, 1994). The accuracy of rule/tree-based taggers is comparable to that of stochastic taggers, yet they are much faster. Moreover, rules or decision trees/lists are human-understandable, thus it can be verified whether or not they capture true underlying linguistic phenomena.

The bulk of the literature on POS tagging is about English. As far as M. Greek is concerned, the primary –to our knowledge– attempt is the stochastic tagger by (Dermatas and Kokkinakis, 1995). They report an error rate of 6% when tagging only with the POS (11 tags), while the error rate increases dramatically (over 20%) when tagging with an extended tag-set (443 tags) that also encodes Number, Case, Person, Tense, etc. In (Orphanos and Tsalidis, 1999) we describe a POS tagger for M. Greek that combines a high-coverage lexicon[1] and a set of decision trees for disambiguation/guessing. This tagger achieves an overall error rate of 7% and assigns full morphosyntactic information to known words while unknown words are being tagged only with their POS[2]. A synopsis of our approach is given in the next section.

THE DECISION TREE APPROACH

When a morphosyntactic lexicon with high coverage is available, the construction of a POS tagger seems a straightforward task. For example, when the words of the following sentence

Ο Βάκης αισθάνθηκε το αίμα του να μυρμηγκιάζει στις κλειδώσεις

are searched in the CTI lexicon, it will return the following tags:

[1] The morphosyntactic lexicon of Computer Technology Institute (CTI) currently contains ~60.000 lemmas (~870.000 word-forms). Given a word-form, the lexicon returns the corresponding lemma (or lemmas in case of lexical ambiguity) along with full morphosyntactic information, i.e. POS, Number, Gender, Case, Person, Tense, Voice, Mood, etc.

[2] A direct comparison of the two taggers for M. Greek is not feasible, since they are trained and tested on different datasets.

1	Ο	Article(Masculine, Singular, Nominative)
2	Βάκης	?
3	αισθάνθηκε	Verb(Singular, Third, Past, Passive, Indicative)
4	το	Article((Singular, Neuter, Nominative | Accusative) | (Singular, Masculine, Accusative)) + Pronoun((Personal, (Singular, Neuter, Nominative | Accusative) | (Singular, Masculine, Accusative))
5	αίμα	Noun(Singular, Neuter, Nominative | Accusative)
6	του	Article(Singular, Masculine | Neuter, Genitive) + Clitic + Pronoun(Personal, Singular, Masculine | Neuter, Genitive) +
7	να	Particle
8	μυρμηγκιάζει	Verb(Singular, Third, Present, Active, Indicative | Subjunctive)
9	στις	PrepositionalArticle(Feminine, Plural, Accusative)
10	κλειδώσεις	Noun(Feminine, Plural, Nominative | Accusative | Vocative) + Verb(Singular, Second, Past, Subjunctive)

Figure 1. An example sentence tagged by the lexicon

One can notice that words #4, #6 and #10 have received two or three tags (words with POS ambiguity), while word #2 has not received any tag since it is not found in the lexicon (unknown word). Also, some words exhibit other-than-POS-ambiguity, e.g. word #2 has Gender/Case ambiguity. Our main aim is to eliminate POS ambiguity for known words and guess the POS of unknown words. The other-than-POS-ambiguity can be resolved later (as well as the guessing of other-than-POS-attributes for unknown words), either by a second disambiguating/guessing layer or by a parser.

According to the tagging performed by the lexicon, a word belonging to *n* POSs receives *n* tags (typically *n* is two or three). Each of the *n* tags contains a different POS value. The goal is to keep the tag with the contextually appropriate POS and discard the rest. On the other hand, the high coverage of the lexicon assures that an unknown word belongs to one of the open-class POSs (i.e., Noun, Verb, Adjective, Adverb or Participle) and therefore the goal is to select the contextually appropriate POS from five possible values, taking also into account the capitalization and the suffix of the unknown word.

The problem of POS ambiguity in its entirety is rather heterogeneous: the decision whether a word is a Noun or a Verb is based on different criteria than the decision whether a word is an Article or a Pronoun. Besides, the Verb-Noun ambiguity cannot be resolved by the same classification device that handles the Article-Pronoun ambiguity, since they pertain completely different classes. Consequently, the entire problem of POS ambiguity must be faced as a set of sub-problems. In order to meet the classification paradigm, all words belonging to a specific sub-problem must receive the same set of POS values. In order to have good classification results, all words belonging to a specific sub-problem must have similar behavior. Taking into consideration these statements, we grouped ambiguous words into sets according to the POS ambiguity schemes revealing in M. Greek, e.g., Verb-Noun, Article-Pronoun, Article-Pronoun-Clitic, Pronoun-Preposition, etc.

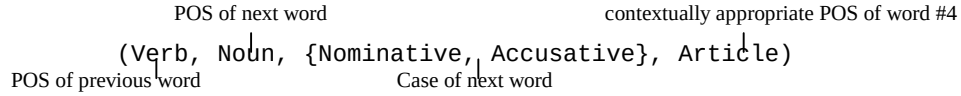
The role of decision trees now becomes evident. The POS disambiguator is, actually, a 'forest' of decision trees, one decision tree for each ambiguity scheme in M. Greek. When a word with two or three tags appears, its ambiguity scheme is identified and the corresponding decision tree is selected. The tree is traversed according to the results of tests performed on contextual tags. This traversal returns the contextually appropriate POS. The ambiguity is resolved by eliminating the tag(s) with different POS than the one returned by the decision tree. Similarly, POS guessing is performed by a decision tree dedicated to this task. When an unknown word appears, its POS is guessed by traversing the decision tree for unknown words, which examines contextual features, the suffix and the capitalization of the word and returns one of the open-class POSs.

We have already said that decision trees examine contextual information in order to carry out the POS disambiguation/guessing tasks. The question that automatically arises is: what sort of tests are performed over the context of an ambiguous/unknown word? The answer is designated by the linguistic problems we try to model: each decision tree examines those pieces of linguistic information that are relative to the decision it has to carry out; the same pieces of information that a human would examine, if it was up to him to decide. Typical tests are: "What is the POS of the previous word?", "What is the Gender of the next word?", "Is the next token a punctuation mark?", etc. It is important to mention that tests do not refer to entire tags but to specific attributes encoded in the tags, a fact that assigns a very significant property to the disambiguating/guessing procedure, namely **tag-set independence**: the lexicon assigns to each known word one or more tags that encode the maximum morphosyntactic information found and the decision trees extract from the tags as much information as they need.

An inherent difficulty of the above arrangement is that a test may result to more than one attribute-values. For example, consider that we have to disambiguate word #4 in Figure 1, which belongs to the Article-Pronoun

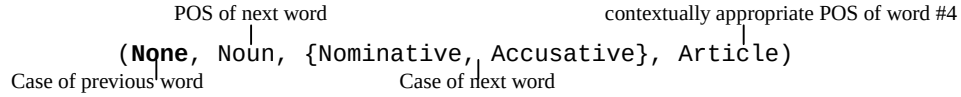
ambiguity scheme. If the decision tree for the Article-Pronoun ambiguity is a generalized[3] tree, one of its nodes might ask: "What is the Case of next word?". It would receive the answer "Nominative or Accusative". This means that there are two possible branches to follow, one starting from the value "Nominative" and one starting from the value "Accusative". A fair policy is to follow the most probable branch, that is to pick the subtree that gathered the greatest number of training patterns. If we had a binary decision tree, such problem would not have occurred during classification, because nodes of these trees ask yes/no questions like: "Is the Case of the Next word Nominative?", "Is the Case of the Next word Accusative?".

The issue of set-valued attributes is not met only during classification, it is also met during learning. Assume that we want to form a training pattern for the Article-Pronoun ambiguity scheme using the example of word #4 in Figure 1 and that the decision tree we want to construct will perform three tests: (a) "POS of previous word", (b) "POS of next word" and (c) "Case of next word". The training pattern would look like:



Although we could eliminate the Case ambiguity, we prefer not to, based on the argument (or the intuition) that the tree must be induced from ambiguous patterns, since later it will have to classify ambiguous patterns. Of course this imposes an extra requirement: the tree induction algorithms should be capable of handling set-valued attributes, regardless of whether they produce generalized trees or binary trees.

A last issue pertains missing values. For example, consider that instead of the test "POS of previous word", we want our tree to perform the test "Case of previous word". Now, the training pattern would look like:



The same would have happened if the tree had to decide about the POS of word #4 and had asked: "What is the Case of previous word?". The answer is "None". "None" during classification could mean "no branch to follow, stop searching and return the default class of the current node". However, this is not exactly the behavior that we expected to achieve. "None" in our example means that the previous word does not have a Case attribute, simply because it is a Verb. In another example, where the ambiguous word might be the first in the sentence, any test relative to its previous token would return "None". Thus, "None" is a meaningful value denoting "I do not have the attribute that you ask. You should proceed to the next test". To be able to capture this behavior, we added an extra value to each test-attribute, the value "None", e.g.:

```
Case = {Nominative, Genitive, Accusative, Vocative, None}
```

DECISION TREE INDUCTION

Decision trees have long been considered as one of the most practical and straightforward approaches to classification (Breiman *et al.*, 1984; Quinlan, 1986). Strictly speaking, induction of decision trees is a method that generates approximations to discrete-valued functions and has been shown, experimentally, to provide robust performance in the presence of noise. Moreover, decision trees can be easily transformed to rules that are comprehensible by people.

There is a couple of very good reasons why decision trees are good candidates for NLP problems, from the classification point of view and especially for POS tagging:

- Decision trees are ideally suited for symbolic values, which is the case for NLP problems.
- Disjunctive expressions are usually employed to capture POS tagging rules. By using decision trees such expressions can still be “discovered” and be associated with relevant linguistic features (note, that the linguistic bias inherent in the representation may also serve as an encoding of produced rules).

Decision trees are built top-down. One selects a particular attribute of the instances available at a node, and splits those instances to children nodes according to the value each instance has for the specific attribute. This process continues recursively until no more splitting along any path is possible, or until some splitting termination criteria are met. After splitting has ceased, it is sometimes an option to prune the decision tree (by turning some internal nodes to leaves) to hopefully increase its expected accuracy.

[3] In a generalized decision tree a node has at the maximum as many children as the different values of the attribute it tests, provided that these values appear during training.

The splitting process requires some effort to come up with informative attribute tests. This paper relaxes the classical definition of the value of an attribute and allows an instance to have a set of values for some attribute. As presented earlier, this deviation is absolutely critical for the POS tagging task.

Set-valued attributes require extra care in how they are handled, as the usual splitting criteria may have to be modified. Specifically, when instances, during training are allowed to follow more than one branch out of a node, it may turn out that the usual entropy-based metrics deliver loss rather gain of information. Needless to say this requires exceptional handling. One of the presented algorithms (algorithm 3) employs a novel pre-pruning strategy for limiting tree growth.

We now give a brief description of the algorithms used in our experiments.

Algorithm 1

Algorithm 1 creates generalized decision trees and uses the gain ratio[4] metric for splitting. Tree growing stops when all instances belong to the same class or no attribute is left for splitting. When an instance, being at a specific node, contains a set of values for the attribute tested by the node, it is directed to *all* branches headed by these values. Each node contains a default class label, which represents the most frequent class of the instances acquired by the node. During a second pass, a compaction procedure eliminates, from the leaves to the root, all children nodes that have the same default class with their father, resulting to smaller trees with identical classification performance.

Algorithm 2

Algorithm 2 is similar to algorithm 1, except that test-attributes are ordered a priori according to their gain ratio measured on the entire instance base. The first split is performed with the first attribute (with the highest gain ratio)and all nodes at level k of the tree test the k^{th} best attribute.

Algorithm 3

Algorithm 3 uses the information gain metric for splitting. It creates binary decision trees. Tree growing stops either when no attribute can differentiate between the instances at a node or when a particular node delivers to (at least) one of its children the whole instance set. Note that this condition can arise when, due to set-valued attributes, instances are directed to both branches. The trade-off for this pre-pruning strategy is that even though one, strictly, observes information loss, it turns out that a repeating pattern of filtering down a path delivers a better accuracy. We have quantified this trade-off by using a *pruning level* parameter. This states, for an instance set, for how many consecutive nodes along a path it may be propagated as is due to imperfect splitting. During testing, an instance, that for a particular attribute has more than one value, will follow more than one path if it arrives at a node that tests the particular attribute. Obviously, it ends up in more than one leaf; its class assignment is the most frequently observed class over *all* reached leaves.

EXPERIMENTATION

Datasets

For the study and resolution of lexical ambiguity in M. Greek, we set up a corpus of 137.765 tokens (7.624 sentences), collecting sentences from student writings, literature, newspapers, and technical, financial and sports magazines. Subsequently, we tokenized the corpus and let the lexicon assign morphosyntactic tags to word-tokens. We did not use any specific tag-set; instead, we let the lexicon assign to each known word all morphosyntactic attributes available. An example of a sentence tagged by the lexicon is already given in Figure 1. Unknown words were tagged with a disjunct of open-class POSs. During a second phase, words with POS ambiguity and unknown words were manually assigned their appropriate POS. Moreover, to unknown words we manually added an attribute representing their suffix.

During the manual disambiguation, we carefully recorded the criteria according which the experts were selecting the contextually appropriate POS. That is to say, for each ambiguity scheme we recorded a set of contextual attributes that assisted the task of manual disambiguation. As expected, different ambiguity schemes require different sets of contextual attributes. Accordingly, we selected from the corpus all instances of ambiguous/unknown words, grouped them into ambiguity schemes and formed training patterns for each ambiguity scheme. The training patterns of an ambiguity scheme encode the contextual attributes relevant to the specific scheme. Thus we succeeded to inject **linguistic bias** to the learning procedure and thus achieve a better approximation to the linguistic problems we try to solve.

A detailed description of the datasets is given in Table 1.

[4] Gain ratio is used instead of information gain, since not all attributes have the same number of values and, as known, information gain favors the most populated attributes.

POS Ambiguity Schemes	Example words	# of instances in the dataset	% occurrence in the corpus
Pronoun-Article	το, τον, τη, την, τις	8500	7,13
Pronoun-Article-Clitic	του, της, τους	5609	4,70
Pronoun-Preposition	με, σε	2551	2,14
Adjective-Adverb	πολύ, λίγο, άδικα, συχνά, βιαστικά, βέβαια, φθηνά, ...	1834	1,53
Pronoun-Clitic	μου, σου, μας, σας	1686	1,41
Preposition-Particle-Conj.	για	1216	1,02
Verb-Noun	αναλύσεις, πράξεις, δηλώσεις, γέννα, πάτε, δίψα, ...	622	0,52
Adjective-Adverb-Noun	μέσα, περιοδικά, καλά, άδεια, λεπτά, ελληνικά, ...	608	0,51
Adjective-Noun	επίπεδο, πολιτική, περιοδικό, τεχνική, μουσική, ...	550	0,46
Particle-Conjunction	δε, δεν, μη, μην	468	0,39
Adverb-Conjunction	πως, πριν, καθώς	429	0,36
Pronoun-Adverb	όσο, μόνο, τόσο	410	0,34
Verb-Adverb	εντάξει, δω, σιγά	82	0,06
	Total POS ambiguity:	**24565**	**20,57**
Unknown Words	σιδερόπετρα, ασβεστόχωμα, πριμοδοτούνται, Πετσάρσκι, ...	**3015**	**2,53**

Table 1. Datasets

Evaluation

To evaluate our approach, we first partitioned the datasets into training and test sets to use 10-fold cross-validation. In this method, a dataset is partitioned 10 times into 90% training material and 10% testing material. The average accuracy over those 10 experiments provides a reliable estimate of the generalization accuracy.

Table 2 illustrates the evaluation results. Column (1) shows the % contribution of each ambiguity scheme to the total POS ambiguity. Column (2) shows the results of a naïve method that resolves the ambiguity assigning the most frequent POS. Column (3) shows the results of algorithm 1. Column (4) shows the results of algorithm 2. Column (5) shows the results of algorithm 3 for pruning level parameters 1, 2, 3 and 4.

POS Ambiguity Schemes	(1) % contribution to POS ambiguity	(2) % error most frequent POS	(3) % error algorithm 1	(4) % error algorithm 2	(5) % error, algorithm 3 pruning levels			
					1	2	3	4
Pronoun-Article	34,6	14,5	1,96	1,96	0,76	0,78	0,73	0,73
Pronoun-Article-Clitic	22,9	39,1	7,43	4,52	5,78	4,41	4,33	4,33
Pronoun-Preposition	10,4	12,2	1,35	1,35	0,39	0,39	0,39	0,39
Adjective-Adverb	7,4	31,1	14,0	13,4	13,05	12,01	11,73	11,80
Pronoun-Clitic	6,8	38,0	6,03	5,78	6,46	5,03	4,96	4,96
Preposition-Particle-Conj.	4,9	20,8	8,94	8,94	7,73	7,73	7,73	7,73
Verb-Noun	2,6	12,1	8,82	10,1	7,70	7,70	7,91	7,70
Adjective-Adverb-Noun	2,4	51,0	31,5	30,4	38,03	27,64	25,09	23,72
Adjective-Noun	2,3	38,2	18,2	20,8	34,54	21,36	19,55	19,55
Particle-Conjunction	1,9	1,38	1,77	1,38	2,89	2,89	3,15	3,15
Adverb-Conjunction	1,7	22,8	23,4	18,1	23,94	23,93	24,54	24,84
Pronoun-Adverb	1,6	4,31	4,81	4,31	5,15	6,12	6,12	6,12
Verb-Adverb	0,4	16,8	1,99	1,99	16,66	3,33	3,33	3,33
Total POS Ambiguity		**24,1**	**7,38**	**6,44**	**6,02**	**4,98**	**4,84**	**4,81**
Unknown Words		**38,6**	**17,8**	**15,8**	**12,29**	**12,55**	**12,46**	**12,33**

Table 2. Evaluation Results

DISCUSSION

We have outlined the use of set-valued attributes in decision tree induction in a linguistic context. This has been possible with relatively straightforward conceptual extensions to the basic model.

A few comments are in order here. By observing the overall behavior of all algorithms over all data sets (precisely, the weighed overall behavior) it is apparent that all decisions tree algorithms provide a significant improvement over the naive heuristic of assigning the most frequent POS.

This dramatic improvement to the naive heuristic and also to the baseline performance by (Dermatas and Kokkinakis, 1995) serve to show that decision trees may well be the solution to the problems of POS disambiguation/guessing in M. Greek.

However, there exist a few discrepancies between the algorithms themselves. Algorithm 3 demonstrates a superior overall performance. The fact that in the latter four data sets it is under-performing is a clear indication of the fact that in the other, most important, cases of POS tagging its superiority is more evident.

There is a very subtle differentiation in the performance of the presented algorithms, which can be best viewed from an evolutionary point of view. First, note, that even though algorithms 1 and 2 utilize the gain ratio metric, they underperform algorithm 3, which uses information gain (which is not usually the case). This leads quickly to the ascertainment of the widely held view that the splitting criterion per se is not of such big importance, when it satisfies some basic quality requirements.

What is very interesting is that algorithm 1 employs a conventional decision trees approach, re-evaluating each attribute's worth in non-root nodes, while algorithm 2 uses the rather unconventional practice of fixing a priority of attribute testing at the root and adhering to it throughout. A close inspection of the tree nodes shows why this might happen. The data set gets excessively fragmented near the tree fringe and splitting tests are based on small samples. This statistical problem is endemic in algorithm 1 whereas algorithm 2 is not subject to it.

Algorithm 3, on the other hand, employs the conventional splitting approach of algorithm 1, but as it may direct instances to more than one path (both during training and during testing), it essentially enlarges the samples on which splitting decisions are based. The size of samples also is reduced at a slower rate than in algorithms 1 and 2, because algorithm 3 implements binary rather generalized decision trees. It may be seen as a moving in parallel with algorithms 1 and 2, utilizing the best features of each one and, finally, overperforming both.

As expected, algorithm 3 is also sensitive to the pruning level. It seems to be the case that the larger the pruning level the better the accuracy. This is, however, not something that can be attributed to the pruning level solely as this behavior does not seem to be uniform over all the experiments. Abnormalities could be safely attributed to the fact that the pruning level heuristic does not employ a quantitative measure of information loss. Its rule of stopping the splitting process is more of a qualitative nature.

We firmly believe that all algorithms will greatly benefit by enhancing them with a suitable post pruning strategy. In particular algorithms 1 and 2 could display a significant performance enhancement. In algorithm 3, performance enhancement may be less evident per se, but we expect it to demonstrate a more orderly behavior regarding its sensitivity to the pruning level. Those items are obviously high on our research agenda.

REFERENCES

Bahl, L. and Mercer, R. (1976) Part-of-speech assignment by a statistical decision algorithm. *International Symposium on Information Theory*, Ronneby, Sweden.

Breiman, L., Friedman, J.H., Olshen, R.A. and Stone C.J. (1984) Classification and Regression Trees. Wadsworth, Belmont, CA.

Brill, E. (1995) Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. *Computational Linguistics*, 21:4, pp. 543-565.

Church, K. (1988) A Stochastic parts program and noun phrase parser for unrestricted text. *Proceedings of 2nd Conference on Applied Natural Language Processing*. Austin, Texas.

Daelemans, W., Van den Bosch, A. and Weijters, A. (1997) Empirical Learning of Natural Language Processing Tasks. In W. Daelemans, A. Van den Bosch, and A. Weijters (eds.) *Workshop Notes of the ECML/Mlnet Workshop on Empirical Learning of Natural Language Processing Tasks*, Prague, pp.1-10.

Daelemans, W., Zavrel, J., Berck, P., and Gillis, S. (1996) MBT: A memory-based part of speech tagger generator. In E. Ejerhed and I. Dagan (eds.), *Proceedings of 4th Workshop on Very Large Corpora*, ACL SIGDAT, pp. 14-27.

Dermatas E. and Kokkinakis G. (1995) Automatic Stochastic Tagging of Natural Language Texts, *Computational Linguistics*, 21:2, pp. 137-163.

Derouault, A. and Merialdo, B. (1984) Language modeling at the syntactic level. *Proceedings of the 7th International Conference on Pattern Recognition*.

Greene, B., and Rubin, G. (1971) Automated grammatical tagging of English. Department of Linguistics, Brown University.

Hindle, D. (1989) Acquiring disambiguation rules from text. *Proceedings of ACL'89*.

Magerman, D. (1995) Statistical decision tree models for parsing. *Proceedings of ACL '95*.

Orphanos, G and Tsalidis C. (1999) Combining Handcrafted and Corpus-Acquired Lexical Knowledge into a Morphosyntactic Tagger. *Proceedings of the 2nd CLUK Research Colloquium*, Essex, UK

Quinlan, J.R. (1986) Induction of Decision Trees, *Machine Learning*, 1:81-106.

Schmid, H. (1994) Part-of-Speech Tagging with Neural Networks. *Proceedings of COLING '94*.

Schmid, H. (1994b) Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of the International Conference on New Methods in Language Processing*., NeMLaP, Manchester, UK

Voutilainen, A. (1995) A syntax-based part-of-speech analyser. *Proceedings of EACL'95*.

Yarowsky, D. (1994) Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French. *Proceedings of ACL '94*.