

# Ensemble Learning for Low Resources Prepositional Phrase Attachment

Pavlos Nalmpantis\*, Romanos Kalamatianos, Konstantinos Kordas, Katia Kermanidis

Department of Informatics, Ionian University, Corfu, 49100, Greece

**Abstract** Prepositional phrase attachment is a major disambiguation problem when it's about parsing natural language, for many languages. In this paper a low resources policy is proposed using supervised machine learning algorithms in order to resolve the disambiguation problem of prepositional phrase attachment in Modern Greek. It is a first attempt to resolve prepositional phrase attachment in Modern Greek, without using sophisticated syntactic annotation and semantic resources, but by employing sophisticated learning techniques i.e ensembles of classifiers.

**Keywords** Decision Trees, Modern Greek, PP attachment, Supervised learning

## 1. Introduction

The correct attachment of prepositional phrases (PPs) to another constituent in a sentence is a significant disambiguation problem for parsing natural languages. For example, take the following two sentences:

1 She eats soup with a spoon.

2 She eats soup with tomatoes.

In sentence 1, the PP "with a spoon" is attached to the verb phrase (VP) "eats", denoting the instrument utilized for the eating action, thus making it the anchor phrase of the PP. Sentence 2 seems to differ only minimally from the first sentence, but, as can be seen from their syntax trees in Fig. 1 and Fig. 2 respectively, their syntactic structure is quite different. The PP "with tomatoes" does not attach to the verb but to the noun phrase (NP) "soup" denoting the type of soup.

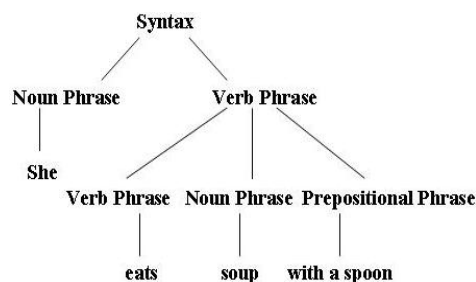


Figure 1. Syntax tree of sentence 1

PP attachment has many significant uses. It can be used, as referred above, for improving the performance of syntactic parsers, as it is a major source of ambiguity in natural language. It facilitates further semantic processing, and also

constitutes an important pre-processing step in many information extraction systems. It has also been employed in speech processing as a filter in prosodic phrasing[1].

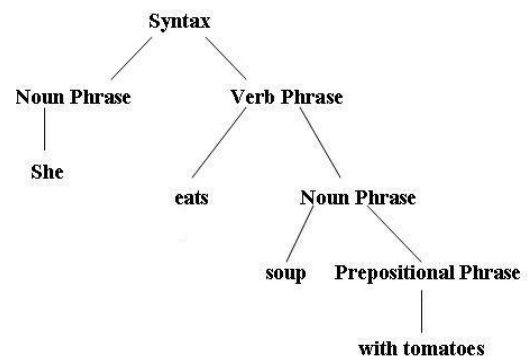


Figure 2. Syntax tree of sentence 2

Over the years, many solutions have been proposed to resolve the disambiguation problem of PP attachment. Solutions include the use of machine learning algorithms[2-3], statistical analysis using corpus-based pattern distributions and lexical signatures[4], as well as the back-off model[5], the maximum entropy model[6] etc. However, these methods usually require many resources (i.e. syntactic annotation and often even semantic disambiguation) which are often unavailable for many languages.

Regarding machine learning techniques, previous approaches have experimented with various learning schemata. Memory-based learning has been proposed[2], employing the 1-NN algorithm (IB1) and its tree-variation (IB1-IG). These results were compared to results of other methods and correct attachment performance turned out much better. Also, a nearest-neighbor algorithm has been proposed, employing a cosine similarity measure of pointwise mutual information [3]. The work described in[7] recreates the EDTBL (Error-Driven Transformation-Based Learning) experiment and compares the results with three machine learning algorithms,

\* Corresponding author:

paylnalm@ionio.gr (Pavlos Nalmpantis)

Published online at <http://journal.sapub.org/ijis>

Copyright © 2012 Scientific & Academic Publishing. All Rights Reserved

namely: Naïve Bayes, ID3 IG (Information Gain) and ID3 GR (Gain Ratio). The latter two are a tree variation of the k-NN algorithm. The experiment was conducted in two phases. In phase 1 the training set was gradually increased until all training examples were used. In phase 2 10-fold cross validation was used, and the summary of results for all the aforementioned approaches is shown in Table 1.

In this paper we propose a methodology to resolve the disambiguation problem of PP attachment in Modern Greek using supervised machine learning algorithms given a dataset of feature-vectors extracted from a morphologically annotated corpus. The presented methodology is, to the authors' knowledge, a first attempt to resolve the PP attachment problem in Modern Greek using minimal linguistic resources. In other words, no sophisticated tools - like grammars, tools, treebanks and semantic thesauri - are utilized. Regarding the machine learning techniques employed, a contribution of the present work is the comparison of the performance of ensembles of classifiers to stand-alone learning algorithms on the task at hand. Finally no restrictions are imposed on the type of the targeted prepositional phrases, i.e. all prepositions are taken into account.

**Table 1.** Summary of Results of Related Previous Work

Algorithm	Results	
IB1(1-NN) [2]	83.7%	
IB-IG(1-NN tree) [2]	84.1%	
Cosine Similarity (NN) [3]	86.5%	
ID3 IG(k-NN) [7]	79%(phase 1)	79%(phase 2)
ID3 GR(k-NN) [7]	78%(phase 1)	79%(phase 2)
Naïve Bayes [7]	74%(phase 1)	76%(phase 2)

The rest of this paper is organized as follows. Section 2 introduces the properties of the Modern Greek language. The corpus used in the presented experiments, the feature extraction process, as well as an example of the creation of the learning vector is discussed in Section 3. Section 4 describes in detail the experimental process and shows the results that were achieved in the experiments. The results are discussed quantitatively and qualitatively in Section 5, and future research prospects are proposed. Finally, the paper concludes with some interesting comments and remarks.

## 2. Modern Greek Properties

Modern Greek is a morphologically rich language, with a complex inflectional system. This leads to a significant degree of freedom in the ordering of the elements of a sentence. The syntactic and semantic roles of the constituents of a sentence are primarily determined by their morphology, rather than their position in the sentence. This phenomenon makes the task of PP attachment in Modern Greek even more challenging. Also internal phrase structure is stricter. Within the noun phrase, adjectives usually precede the noun (e.g. “το μεγάλο σπίτι”, [to megalo spiti], ‘the big house’), while possessors follow it (e.g. “το σπίτι μου”, [to spiti mu], ‘my house’). Regarding verb phrases, certain grammatical ele-

ments attach to the verb as clitics and form a rigidly ordered group together with it. This applies particularly to unstressed object pronouns, negation particles, the tense particle “θα” [θa], and the subjunctive particle “να” [na][8].

Prepositional phrases in Modern Greek are introduced by one of the following 21 prepositions[9]:

1. από - from
2. σε - in
3. για - for
4. με - with
5. προς - to
6. παρά - despite
7. σαν - as
8. ως - as
9. μέχρι - until
10. χωρίς - without
11. δίχως - without
12. μετά - with
13. υπέρ - over
14. εναντίον - against
15. εξαιτίας - because
16. λόγω - because
17. μεταξύ - between
18. αντί - instead
19. κατά - at
20. πριν - before
21. ίσαμε - up to

They may constitute direct or indirect objects to verbs as modifiers of the other constituents. The preposition “σε”, when followed by a definite article, fuses with it into forms like “στο” (σε + το) and “στη” (σε + τη). For this reason lemmatization of the preposition is required.[8]

## 3. Modern Greek PP-Attachment

### 3.1. Corpus and Pre-processing

The text corpus used in the experiments is the ILSP/ELEFTherOTYPIA corpus[10]. It consists of 5244 sentences; it is balanced in domain and genre, and manually annotated with complete morphological information. Further (phrase structure) information is obtained automatically by a multi-pass chunker[11].

During chunking, NPs, VPs, PPs, adverbial phrases (ADP) and conjunctions (CON) are detected via multi-pass parsing. The chunker exploits minimal linguistic resources: a keyword lexicon containing 450 keywords (i.e. closed-class words such as articles, prepositions etc.) and a suffix lexicon of 300 of the most common word suffixes in Modern Greek. The chunked phrases are non-overlapping. Embedded phrases are flatly split into distinct phrases. Nominal modifiers in the genitive case are included in the same phrase with the noun they modify; base nouns joined by a coordinating conjunction are grouped into one phrase. The chunker identifies basic phrase constructions during the first passes (e.g. adjective-nouns, article-nouns), and combines smaller

phrases into longer ones in later passes (e.g. coordination, inclusion of genitive modifiers, compound phrases).

### 3.2. Feature Selection

The feature vector is necessary to enable the classification algorithms to classify the PP attachment. The attributes which can be used to form the feature vector need to be related to the PP attachment ambiguity resolution task, and they vary in the bibliography. The ones encountered most commonly in previous work (e.g.[12]) are: the number of commas between the PP and the anchor candidate, the number of other punctuation marks between the PP and the anchor candidate, the number of words between the PP and the anchor candidate, the POS-tag of the last token of the anchor candidate, the lemma of the PP, the number of PPs between the PP and the anchor candidate, the label of the phrase immediately before the PP, the anchor candidate, etc.

In the present work, a feature vector is formed for every anchor candidate and every preposition in a given corpus sentence. Thus, the syntactic freedom of Modern Greek is taken into account. Of the features we mentioned earlier, after a number of experiments conducted using the tools of the WEKA machine learning workbench (Waikato Environment for Knowledge Analysis)[13], which enables the experimentation with various classification algorithms, the following features were selected to form our feature vector:

- The lemma of the preposition introducing the PP.
- The type of the phrase immediately before the PP.
- The anchor candidate.
- The POS-tag of the last token of the anchor candidate.
- The number of words between the PP and the anchor candidate.
- The number of PPs between the PP and the anchor candidate.
- The number of commas between the PP and the anchor candidate.
- The number of other punctuation marks between the PP and the anchor candidate.

The feature “Correct attachment” was used as the classification class of the vector in the experiments conducted. The reason we selected the feature “POS-tag of the last token of the anchor candidate” is that, in most cases, the headword of NPs and the main verb in VPs is the last token of the phrase. The anchor candidate may be preceding or following the PP, as the syntactic freedom of the language does not impose restrictions on the ordering. Therefore, no such restrictions have been imposed on the described feature set.

### 3.3. Feature Vector Extraction

The process of exporting values for each feature vector is automated. We created a program that is written in C language that automatically identifies the first eight features of our vector and stores the results in an Excel file. This program gives all possible attachments (anchor candidates) of a PP in a sentence, that are NPs and VPs, as these phrase types constitute the most significant error source for PP attachment.

However it is fairly easy, in future research, to include other anchor candidate phrase types. The correct class label was assigned to every extracted feature vector by three language experts, manually. This feature takes the values of TRUE or FALSE and indicates whether the attachment example represented in the given vector is correct or not. Inter-annotation agreement between the experts was 90%. For the remaining 10%, where the experts didn't initially agree, a discussion among them followed, resulting to a common decision about correct attachment.

An example of the feature vector extraction process follows. Take the following annotated sentence (firstly presented in the Modern Greek language and then translated into English).

Modern Greek:

NP[Διάλογος<N\*διάλογος> σύγχρονου<A\*σύγχρονος> εργαζόμενου<N\*εργαζόμενος>] PP[με<S\*με> τη<T\*ο> \*σύζυγο<N\*σύζυγος>] ADP[χθες<R\*χθες>] NP[το<T\*ο> \*μεσημέρι<N\*μεσημέρι>.<F>]

English:

NP[The dialogue of a contemporary worker] PP[with his wife] ADP[yesterday] NP[afternoon.]

Each word in the sentence is annotated with its POS-tag and lemma. Table 2 shows the value of each symbol that represents the POS-tag of a word.

The program will extract two feature vectors, which are presented in Table 3, giving us two possible attachments: The PP “with his wife” could be attached to the NP “The dialogue of a contemporary worker” or to the NP “afternoon”. These two possible attachments will be examined by the three language experts mentioned earlier and they will decide upon the correct attachment. In this case the PP “with his wife” (“με την σύζυγο” in Modern Greek) is correctly attached to the NP “The dialogue of a contemporary worker” because it denotes the person with whom the dialogue took place.

**Table 2.** Pos-Tag Symbols

Symbol	Value
N	Noun
A	Adjective
S	Preposition
T	Article
R	Adverb
F	Punctuation mark

**Table 3.** Feature Vector

L	Pre-P	AC	POS-tag	#W	#PPs	#C	#PM
με	NP	NP	N	0	0	0	0
με	NP	NP	F	1	0	0	0

The explanation of the symbols in Table 3 follows next: L=Lemma, Pre-P=Previous Phrase, AC=Anchor Candidate, POS-tag=Anchor Candidate last word, #W=word distance between PP and anchor candidate, #PPs=number of PP's between PP and anchor candidate, #C=number of commas, #PM=other punctuation marks.

## 4. Machine Learning Algorithms

In this section all machine learning algorithms that were used in the experiments, are described.

J48[14] is a decision tree induction algorithm and it is a version of C4.5, an earlier algorithm developed by J. Ross Quinlan[17]. C4.5 creates a decision tree based on the attribute values of the available training data. Whenever it encounters a set of items (training instances), it identifies the attribute that discriminates them most clearly according to their class label. Additionally, J48 incorporates two tree pruning methodologies: The first one is known as subtree replacement and it replaces a node in a decision tree and its subtree with the corresponding leaf, if the given subtree does not help classification accuracy. The process of this type of pruning starts from the leaves of the fully formed tree and moves bottom up toward the root. The second methodology is known as subtree raising in which a node may replace other nodes while it is moved towards the root. This type of pruning most of the times has insignificant effect on decision tree models.

The IBk algorithm[14] is an alternate version of the *k*-nearest neighbor algorithm and uses the same distance metric. *k* is an integer that represents the number of the neighbors. *k* can be defined explicitly or can be specified automatically using leave-one-out cross-validation. When *k* is bigger than 1 ( $k > 1$ ), meaning that the number of the nearest neighbors taken into account is more than one, then the distance between the neighbors is converting into a weight. In this case majority voting is used to classify the unseen instance. Also when new training instances are being saved by the classifier the previous ones are removed so as to keep the same size of training instances.

Naïve Bayes[14] is a probabilistic classifier based on the assumption of conditional independence[18], which assumes that the appearance of a specific feature given the class value is unrelated to the appearance of any other feature. It is based on Bayes' theorem with strong independence assumptions. The naïve Bayes algorithm may be characterized as an "independent feature model". Additionally, this algorithm needs a small amount of training data to determine the parameters necessary for classification. Due to the hypothesis of independent variables; there is no need to estimate the entire covariance matrix but only the differentiations of the variables for each class.

Sequential Minimal Optimization or SMO[15] is an ameliorated algorithm for training support vector machines. SMO cuts in pieces a large quadratic programming optimization problem converting it into smaller problems (sub-problems of quadratic programming). The sub-problems are solved quickly because they are solved analytically which means that SMO avoids to use extra time for arithmetical quadratic programming optimization as an inner loop. So SMO manages to reduce computation time significantly.

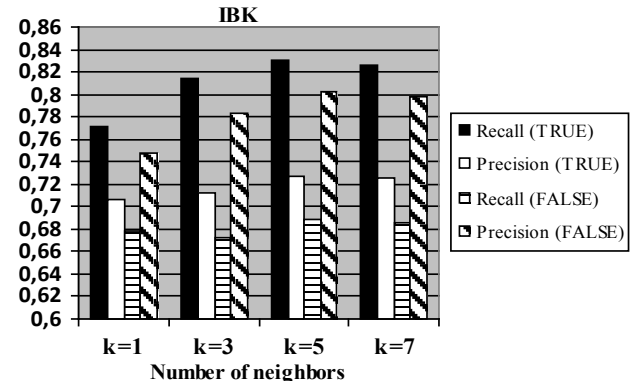
Stacking[14] is used in the present work for the first time to PP attachment. Stacking combines multiple models produced by different learning algorithms (base learners).

Stacking tries to find out which base learner gives the most reliable results. In order to achieve this, stacking uses another learning algorithm, the metalearner, to detect the best way to pair the output of the base classifiers. The input to the model that is going to be created by the Stacking algorithm is the classification results of the base models.

## 5. Experimental Process

Machine learning is used in order to train the system to be able to make "smart" decisions regarding the anchor candidate phrase to which the PP phrase is to be attached. The produced dataset consisted of 8500 vectors corresponding to 500 corpus sentences. Consequently, the type of machine learning we used is supervised machine learning and, having already observed the right results-outputs which have been given by the experts, the system will be able to predict results automatically.

Of the 8500 vectors, only 7.9% of them indicated a correct attachment (positive examples) and the remaining 82.1% indicated an erroneous attachment (negative examples), so we had an imbalanced dataset and biased results (recall and precision were much higher for the negative class than for the positive class). To balance the dataset random under-sampling[16] was performed, i.e. the random removal of negative instances in order for the remaining to reach the number of positive instances. The final number of instances was 1350.

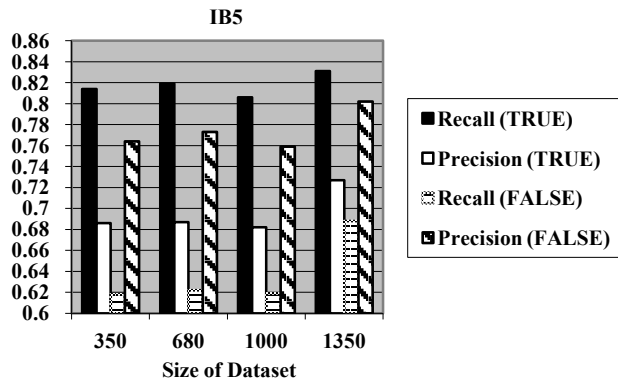


**Figure 3.** Precision & Recall for both class labels as a function of the number of nearest neighbors (value of *k*) using ibk 1350 vectors

In order to validate our results we used 10-fold cross-validation. Through this method, the original sample is randomly partitioned into ten subsamples. One out of ten subsamples is retained as validation data for testing the model, and the remaining nine subsamples are used as training data. The cross-validation process is then repeated ten times, with each of the ten subsamples being used only once as validation data.

The results of the conducted experiments, i.e. precision and recall for both class labels, are shown in Fig. 3 to 7. IBk was run for various values of *k* (Fig. 3). The results with  $k=5$ , being the highest, are then depicted for various data set sizes in Fig. 4. Fig. 5, Fig. 6 and Fig. 7 show the results with J48

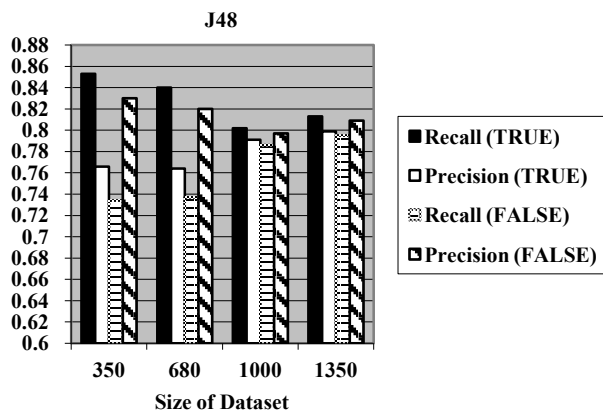
and Naïve Bayes respectively. Values for  $k$  higher than 5 apparently lead to the inclusion of more noise than “clean” data in the decision process.



**Figure 4.** Precision & Recall for both class labels as a function of the dataset size using IB5

## 6. Evaluation

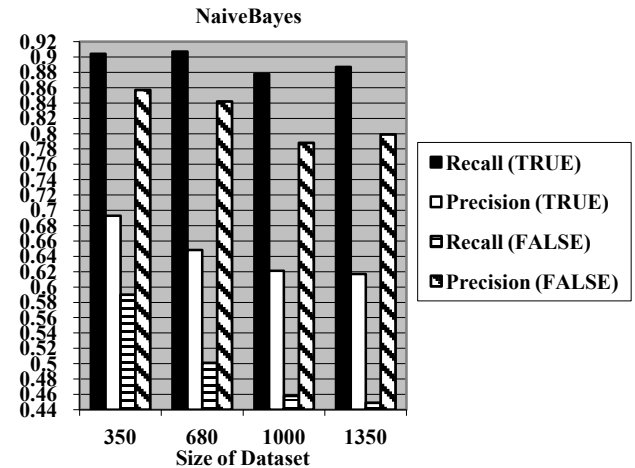
Using the J48 algorithm one may notice that the most significant feature is the “Candidate and PPs word distance”. It is the feature that is placed at the root of the tree constructed by the C4.5 classifier. In other words, it is the first feature to be checked in order to classify a new unseen attachment example. The constructed tree is represented in Fig. 8. Table 4 explains the integers on the decision tree nodes in Fig. 8.



**Figure 5.** Precision & Recall for both class labels as a function of the dataset size using J48

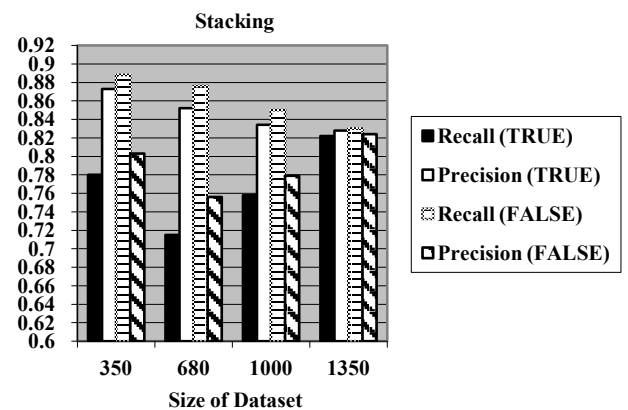
It is clear from the tree structure that the features “Preposition Lemma”, “Pos-Tag Candidates last token” and “PPs between candidate and PP” are leaves of the tree constructed by C4.5 classifier. This means that these features are not so significant. From the representation of the tree (Fig. 8) it is evident that when the feature “candidates and PPs word distance” has a value greater than 3, the example is usually classified as false. Therefore, many examples that should be classified as true, are classified as false, because their value in this feature is great. From the results produced by the J48 algorithm in Fig. 5 one may notice that as the size of the dataset increases, recall and precision of both class labels

start to converge with each other. Using the IBk algorithm we can see in Fig. 3 that as the number of the nearest neighbors increases until getting the value 5 (IB5), recall and precision of both class labels rise. However, at this point, when the number of nearest neighbors takes values greater than 5, the performance of the learner is affected due to noise introduced by the ‘distant’ closest neighbors. Also, from the results that were produced by the IBk algorithm with 5 nearest neighbors (Fig. 4), recall and precision of both class labels rise as the size of the dataset increases.



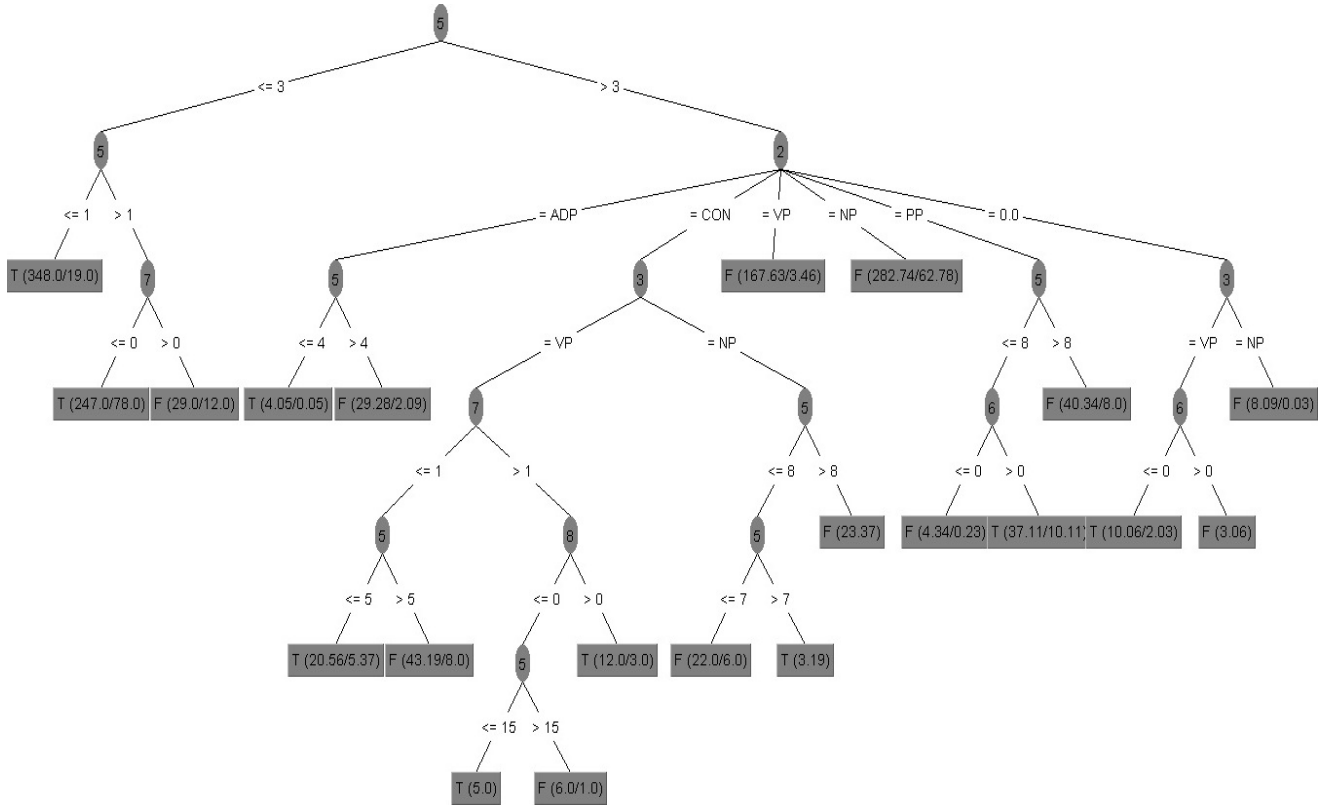
**Figure 6.** Precision & Recall for both class labels as a function of the dataset size using Naïve Bayes

Due to the conditional independence assumption of Naïve Bayes, the results produced by this algorithm are poor compared to other algorithms. From Fig. 6, it is evident that, as the size of the dataset increases, recall and precision of both class labels decrease. Recall of class label TRUE and precision of class label FALSE increase slightly when the size of the dataset consists of 1350 vectors.



**Figure 7.** Precision & Recall for both class labels as a function of the dataset size using Stacking

The best performance (comparing all base learning algorithms and dataset sizes) was achieved with J48 with the 1350 dataset (recall and precision of approximately 80%). Compared to the previous work results in Table 2, the results are comparable to previous approaches that utilize more sophisticated external resources.



**Figure 8.** This figure depicts the tree constructed by the C4.5 classifier. T stands for class label TRUE and F for FALSE

Number	FEATURE
1	Preposition Lemma
2	POS-Tag word before PP
3	POS-Tag Candidate
4	POS-Tag Candidates last token
5	Candidates and PP's word distance
6	PP's between Candidate and PP
7	Commas between PP and Candidate
8	PM between PP and Candidate
9	Correct

Using the Stacking method the best overall results were achieved. After extensive experimentation, the following algorithms were used as base classifiers: J48, Naïve Bayes, IBk(k=5) and SMO algorithms, J48 was also used as the meta classifier. We can see in Fig.7 that as the size of the dataset increases, recall and precision of both class labels start to converge with each other. Precision of class label TRUE and recall of class label FALSE decrease as the size of the dataset increases but these classes slightly decrease when the size of the dataset consists of 1318 vectors. Precision of class label FALSE and recall of class label TRUE get their lowest value as the size of the dataset consists of 774 vectors and then start to increase as the size of the dataset increases.

## 7. Conclusions

Supervised machine learning was used in order to solve

the PP attachment problem in Modern Greek. The proposed policy was a low resources solution employing basic syntactic pre-processing, and no use of sophisticated parsing tools or semantic thesauri was made. Also, no restrictions were placed upon the prepositions that were addressed, i.e. all prepositions were taken into account. Stacking was applied for the first time to the task, with impressive results, compared to stand-alone classifiers.

Future research could take into account other anchor candidate phrase types. Furthermore, the above experiments could be repeated with different learning algorithms and/or a different set of features which could lead to even better results. The low resource policy allows for the easy portability of the presented methodology to other languages that lack in resources, another interesting research directive that could be explored in the future.

## REFERENCES

- [1] O. Van Herwijnen, J. Terken, A. van den Bosch and Erwin Marsi, "Learning PP attachment for filtering prosodic phrasing," in Proceedings of the European Chapter of the Association for Computational Linguistics, Budapest, Hungary, 2003. Available: [http://ilk.uvt.nl/downloads/pub/papers/paper\\_EACL03.pdf](http://ilk.uvt.nl/downloads/pub/papers/paper_EACL03.pdf).
- [2] J. Zavrel, W. Daelemans and J. Veenstra, "Resolving PP attachment ambiguities with memory-based learning," in Proceedings of the Conference on Computational Natural Language Learning, pp. 136-144, 1997.
- [3] S. Zhao and D. Lin, "A nearest-neighbor method for resolving PP-attachment ambiguity," in 1st International Joint Conference on Natural Language Processing, 2004.
- [4] N. Gala and M. Lafourcade, "Combining corpus based pattern distributions with lexical signatures for PP attachment ambiguity resolution," in Proceedings of the 6th Symposium on Natural Language Processing, Chiang Rai, Thailand, 2005..
- [5] M. Collins and J. Brooks, "Prepositional phrase attachment through a backed-off model," in Proceedings of the Third Workshop on Very Large Corpora, pp 27-38, 1995, Cambridge.
- [6] A. Ratnaparkhi, J. Reyna.r and S. Roukos, "A maximum entropy model for prepositional phrase attachment," in Proceedings of the ARPA Workshop on Human Language Technology, pp. 250-255, Plainsboro, N J, March 1994.
- [7] B. Mitchell and R. Gaizauskas, "A comparison of machine learning algorithms for prepositional phrase attachment," in 3rd International Conference on Language Resources and Evaluation, pp 2082-2087, Las Palmas, Canary Islands, Spain, 2002.
- [8] D. Holton, P. Mackridge and I. Philippaki-Warburton, Greek: a comprehensive grammar of the modern language, Routledge, London, 1997.
- [9] Ειρήνη Φυλιππάκη-Warburton, Μιχαήλ Γεοργιαφέντης, Γεώργιος Κοτζόγλου, Μαργαρίτα Λουκά, "ΓΡΑΜΜΑΤΙΚΗ Ε΄ ΚΑΙ ΣΤ΄ ΔΗΜΟΤΙΚΟΥ". Οργανισμός εκδόσεων διδακτικών βιβλίων, Ενότητα 13 Προθέσεις, pp. 131-132
- [10] N. Hatzigeorgiu, M. Gavrilido, S. Piperidis, G. Carayannis, A. Papakostopoulou, A. Spiliotopoulou, A. Vacalopoulou, P. Labropoulou, E. Mantzari, H. Papageorgiou and I. Demiros, "Design and implementation of the online ILSP Greek corpus," in 2nd International Conference on Language Resources and Evaluation, pp. 1737-1742. Athens, Greece (2000). Available: <http://www.elda.fr/catalogue/en/text/W0022.html>.
- [11] E. Stamatatos, N. Fakotakis and G. Kokkinakis, "A practical chunker for unrestricted text," in Conference on Natural Language Processing, pp. 139-150. Patras, Greece, 2000, submitted for publication.
- [12] V. Van Asch and W. Daelemans, "Prepositional phrase attachment in shallow parsing," in 7th International Conference on Recent Advances in Natural Language Processing, Borovets, Bulgaria, 2009.
- [13] <http://www.cs.waikato.ac.nz/ml/weka/>.
- [14] Witten IH, Frank E. "Data Mining: Practical Machine Learning Tools and Techniques". Second edition, 2005. Morgan Kaufmann.
- [15] John C. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines"
- [16] N. V. Chawla, "Data mining for imbalanced datasets: an overview(Periodical style)", Dept. of Computer Science and Engineering, Notre Dame Univ., U.S, 2005.
- [17] J. R. Quinlan, "Bagging, boosting and C4.5," in 13th National Conference on Artificial Intelligence, Portland, Oregon, 1996.
- [18] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd edition, Prentice Hall, 2009.