An alternative approach for statistical single-label document classification of newspaper articles

Article *in* Journal of Information Science · June 2011

DOI: 10.1177/0165551511403543 · Source: DBLP

3 authors:

Georgios Mamakis
Hellenic Mediterranean University
7 PUBLICATIONS **26** CITATIONS
SEE PROFILE

Athanasios G. Malamos
Hellenic Mediterranean University
71 PUBLICATIONS **282** CITATIONS
SEE PROFILE

J. Andrew Ware
University of South Wales
72 PUBLICATIONS **912** CITATIONS
SEE PROFILE

An Alternate Approach on Statistical Single-label Document Classification for Greek Newspaper Articles

Georgios Mamakis
Department of Applied Informatics & Multimedia, Technological Educational Institute of Crete, Greece
Faculty of Advanced Technology, University of Glamorgan, Wales

Athanasios G. Malamos
Department of Applied Informatics & Multimedia, Technological Educational Institute of Crete, Greece

J Andrew Ware
Faculty of Advanced Technology, University of Glamorgan, Wales

Abstract

Text classification is one of the most important sectors of machine learning theory. It enables a series of tasks among which are email spam filtering, text summarization and context identification. It has been one of the most intriguing tasks in the computer theory, as it represents the projection of human thought in machine form. Classification theory proposes a number of different techniques based on statistics (Naive Bayes Classifier (NBC), language models (LM)), artificial intelligence (Neural Networks) and decision trees among others. Classification systems are typically distinguished into single-label categorization and multi-label categorization systems, according to the number of categories they assign to each of the classified documents. In this paper, we present work undertaken in the area of single-label classification which resulted in a statistical classifier, based on the Naive Bayes assumption of statistical independence of word occurrence across a document. Our algorithm, takes into account cross-category word occurrence in deciding the class of a random document. Moreover, instead of estimating word co-occurrence in deciding a class, we estimate word contribution for a document to belong in a class. This approach outperforms other statistical classifiers as Naive Bayes Classifier and Language Models, as it was proven in our results.

Keywords: single-label document classification / categorization, statistics, Naive Bayes Classifier, Language Models

1. INTRODUCTION

Text classification or text categorization is the task of assigning a random document to a class (single-label classification) or a number of classes (multi-label summarization) retrieved from a pre-defined set of possible categories. A special case of single-label classification is binary classification, where the systems choose between two possible classes. Text classification may be applied to numerous areas such as text summarization (single-label classification), email spam filtering (binary classification) and context identification (multi-label classification) among others. Numerous approaches have been proposed to achieve such a task including among others statistics, vector space models, artificial intelligence, decision trees and rule-based methods. One of the simplest approaches is statistical classifiers. The initial assumption of statistical classifiers is the exploitation of observed features that are present in a document, such as word or character occurrence. Statistical classifiers, despite their simplicity or naive assumptions are proven to achieve great results, outperforming in many cases more complex algorithms on a speed-efficiency trade-off (Kotsiantis, Pintelas, 2004). Another positive characteristic is their speed of execution as opposed to other more complex approaches (SVMs) , which allows them to be used for real-time applications (Tsoumakas, Katakis,

Vlahavas, 2006). These are the reasons why we chose to research statistical classifiers in general, as part of the work to be presented is based on statistics and is part of a greater project for real-time corpus management. Work on statistical methods for text classification as NBC has been undertaken by numerous researchers (Rennie, Shih, Teevan, Karger, 2003), (McCallum, Nigam, 1998), (Nigam, McCallum, Thrun, Mitchell, 2000), (Rish I.,2001), (Dai, Xue, Yang, Yu, 2007), (Peng, Schuurmans, Wang, 2004) with significant results regarding its simplicity and efficiency, while statistical learning models have often initiated interest either as an autonomous statistical approach (Srinkanth, Srihari, 2002), (Croft, 2003) , (Ponte, Croft, 1998) alternate to NBC or in conjunction to NBC to achieve better results (Peng et al, 2004). Apart from statistical methods another common approach is the definition of vector space models, utilizing algorithms such as k-Nearest Neighbor (kNN) (Han, Karypis, Kumar, 2001), (Tan, 2005), (Kwon, Lee, 2003) and Rocchio algorithm (Rocchio, 1971), (Joachims, 1997), (Moschitti, 2003), (Miao, Duan, Zhang, Jiao, 2009) These algorithms try to identify the similarities of random documents based on a 2D representation of training data either by approximating similarity based on proximity information of a random document through its pre-classified neighbors (kNN), or by visualizing a 2D space split by lines, planes or hyperplanes, denoting the classes a random document may belong to, and try to fit the document into the most appropriate class based on word similarity through vector distance (Rocchio algorithm). Other approaches include artificial intelligence classifiers (Neural Networks) as in (Zhu, Wang, Yao, Tsou, 2008), (Zhang, Zhu, 2006) and (Ruiz, Srinivasan, 2002), decision trees (C4.5) (Duchrow T., Schtatland T., Guettler D., Pivovarov M., Kramer S., Weissleder R., 2009), (Ruggieri, 2002) and rule-based methods (CN2) (Clark, Niblet, 1989).

A common approach shared by all these classification methods includes a training process, where the corresponding systems use a training example set of words or documents already classified into their according classes, and a test dataset, where the efficiency of the systems is estimated, after the training phase. Online corpora to assist in that direction exist, such as the TREC (2010) and Reuters (2010) corpora for the English language.

In this paper, we deal with single-label, newspaper article classification. We have developed a statistical classifier for Greek language, based on Naive Bayes assumption and cross-category word co-occurrence. In our algorithm, we compute that given a word belonging to a specific category, the probability the document to belong to that category. Instead of assuming co-existence of word (Naive Bayes assumption) we calculate the Expected Value of a random document to exist in a certain category when a specific word occurs in one category. The Expected Value of the document to exist in a certain is based on the fact that words may occur in more than one categories, and is calculated through a statistical weight function. The rest of the paper is organized as follows: in Section 2 we provide insight on statistical classification methods and primarily on Naive Bayes Classifier and Language Models, while in Section 3, we present our approach which significantly differentiates from both NBC and LM. In later sections, we provide the algorithm we developed and results acquired from 1015 greek newspaper articles (as the system is part of a text summarizer for Greek language) and a test corpus of 353 newspaper articles divided in six categories. In the last chapters, we try to evaluate the acquired results and conclude with a few words on the efficiency of our algorithm and future work.

2. BACKGROUND ON STATISTICAL CLASSIFIERS

Single document classifiers have been of utmost importance in the area Machine Learning, mostly due to their ease of use, simplicity and efficiency. Two of the most important statistical algorithms for classification are Naive Bayes Classifier and Language Model. Both of these algorithms try to extract statistically important information form a random document, and through a training process try to fit a random document to one of the accepted categories.

2.1 Naïve Bayes Classifiers

Naive Bayes Classifiers are supervised learning classifiers, based on the Bayes theorem with strong independence assumptions on feature occurrence in a random document. This implies that the occurrence of each feature (word in our case) in a document contributes independently to the potential class of the document. Let a set of classes C and a set of features $X=(x_1, x_2, \ldots, x_n)$. Classification is based on the maximization of P(C|X)

According to Bayes Theorem

$$P(C \mid X) = \frac{P(X \mid C) * P(C)}{P(X)} \quad (1)$$

Since P(X) is the same for any given class in our example set then Formula 1 may be transformed to

$$P(C \mid X) = P(X \mid C) * (P(C) \quad (2)$$

This is analyzed to

$$P(C|X)=P(C)*P(x_1,x_2,..,x_n|C) \quad (3)$$

and since we are referring to independent features x_n in the feature set X, then the final formula for calculating the probability a given document with a set of characteristics X to belong to a category C becomes

$$P(C|X) = P(C)* \prod_{i=1}^{n} P(x_i \mid C) \quad (4)$$

The document, therefore, belongs to the class which maximizes this *a posteriori* probability, often referred to as Maximum a posteriori decision rule (MAP).

Naive Bayes Classifiers have been used extensively in document classification, either as a baseline classifier with which one may compare (almost every paper in the supplied bibliography compares with Naive Bayes Classifier), or through extensions on this initial representation of the algorithm, in order to tackle known problems of the classifier. The strengths of the classifier have been outlined in (Rish, 2001) where the author proved through simulation that NBC perform best on completely independent features, which is expected given the initial hypothesis, and on functionally dependent features. This work also underscores the fact that the algorithms efficiency is lower in between these two extremes. Moreover, the authors in (Rennie et.al., 2003)tried to find inherent problems of Naive Bayes Classifiers and correct them in order to achieve better results. Thus, they found that NBC is bias-prone if the training sets used are uneven and that it suffers from what they refer to as Weight Magnitude Errors; that is words that may be found together e.g. San Francisco vs. Boston getting double score per occurrence due to the fact that San Francisco is comprised of two words rather than one. Extensions on NBC have been proposed by a number of researchers. McCallum and Nigam, for example, in (1998) and (Nigam et.al., 2000) and Dai et al in (2007) proposed several extensions on NBC with Expectation – Maximization algorithms, in order to face the costly task of manually labeling an example corpus, by using a small set of labeled corpus and a large set of unlabeled corpus.

2.2 Language Models

Another statistical approach extensively used in text classification is Language Models. Language Models (Dai et al, 2007), (Ponte, Croft, 1998) are based on word co-occurrence. They evaluate this co-occurrence by assigning a probability to a sequence of words, by computing its probability distribution. When referring to document classification, language model is associated with a document in an example set and the random document is evaluated according to the similarity with the language model. Due to the fact that it is not always possible to evaluate the language model in text corpora due to great number of words that may create the language model an n-gram approach may be followed. Hence, in an n-gram language model, the probability of the observation of a sentence $W=(w_1,w_2, \ldots,w_k)$ can be calculated as

$$P(W)=P(w_1,w_2,\ldots w_k)= \prod_{i=1}^{k} P(w_i \mid w_1, w_2 ,..w_{i-1}) \approx \prod_{i=1}^{k} P(w_i \mid w_{i-(n-1)} ,..w_{i-1}) \quad (5)$$

Since, it is considered that the probability of the occurrence of word I of the sentence in the context history of the preceding words can by approximated by the probability of observing it in the previous n-1 words. Learning models have been used as an alternate approach to NBC in an attempt to evaluate the statistical dependence of words that may be apparent in a sentence. An estimation of the maximum likelihood estimate of the n-gram probabilities may given by the observed frequency

$$P(w_i \mid w_{i-(n-1)} ,..w_{i-1}) = \frac{(w_{i-(n-1)}...w_i)}{(w_{i-(n-1)}...w_{i-1})} \quad (6)$$

One of the first research works in Language Models for IR was undertaken by Ponte and Croft [10], where they proposed a language modeling technique for classification tasks, and carried out experiments that proved that Language Models produced better results than standard tfidf weighting techniques. Examples of work in language models have been undertaken by Dai et al (2007) where the authors tried to enhance Naïve Bayes Classifier with Language Models, in order to overcome the statistical independence of NBC. Language Models infer ordered sequence of words as they appear in a sentence, in order to estimate the statistical dependence of the word sequence occurrence. Language models have been used as a means to estimate unordered word occurrence by Srikanth and Srihari in (2002), where they proposed three approaches on estimating unordered word occurrence (referred to as biterm as it consists of two words) on random documents: through the average of the components of a bigram language model, the term frequencies of both words to the occurrence of the first observed word, and the term frequencies of both words to the minimum of the term frequencies. Their experiments showed that Language Models may fail to be as effective as unordered word n-ples observation.

3. OUR APPROACH

The Naive Bayes assumption of statistical independence states that words in a document are statistically independent with regards to their appearance, and through the Naive Bayes Classifier has been proven to work surprisingly well, considering the initial false assumption. However, NBC classifier suffers from bias over unequal in length classes (as proven by Rish in (2001)). In our approach, we have developed a single-label, newspaper article supervised trainable classifier for Greek language that uses a normalized approach in assigning a random document to a class. The system takes into account term frequency, along with the size of each class. Therefore, each class component has a normalized weight coefficient, participating in the final classification step. The main difference with

common approaches, as the NBC, is that we did not utilize a product methodology in computing the importance of words in a document to extract the category a document belongs to, but rather the sum of each of the weight coefficient of each word for a category. Our approach considers that each word is statistically independent in occurring in a random document, and instead of trying to identify word co-occurrence as in NBC and LMs (computing the product of each probability of word occurrence, therefore searching for word co-occurrence), we estimate the contribution of each independent word to the class of the document. Given a word in a document, we consider the probability of the document to belong to a specific category featuring that word, and we further estimate the expected value (mean probability) of the contribution of the entire word set of the document to each one of the available categories. A direct outcome from using the sum of probabilities rather than the product is that, while we were initially targeting to classify a document in exactly one of the available categories (single-label classification), the system was able to identify more than one potential categories (multi-label classification). Another outcome is that error is not propagated in sum as quickly as in the product, thus making the algorithm less susceptible to noise.

Another assumption we made in developing our system dealt with what words should be considered as important in our system. We consider only nouns to be important in identifying the context of a sentence or document rather than other parts of speech (Galley,McKeown, 2003). This has also been verified in work by Bouras and Tsogkas in (2010), where the authors implied that nouns extraction greatly assists in Classification and Summarization tasks. The reason for that is that we consider the nouns to hold the essence of a sentence, while verbs and other parts of speech operate either complementary to the meaning or show action between nouns. Thus, a sentence for our system is denoted by the stems of the nouns the beginning of a paragraph and a dot/question mark/exclamation mark or two dot/question marks/exclamation marks.

Our initial thoughts formulated the following problem: Let i a random noun, and D a random document featuring word i, and j a category the document may belong to. We are trying to compute what is the possibility for document D to belong to category j given that word i appears in D.
In order to compute this we assign as $w_{i,j}$ as the weight of word i in category j computed as

$$w_{i,j} = \frac{|tf(i,j)|}{\sum_{i=0}^{n} |tf(i,j)|} \tag{7}$$

where *tf (i,j)* the term frequency of word i in category j, and n the total number of unique words comprising category j. $w_{i,j}$ in this context denotes the importance or contribution of noun i in category j. By dividing with the total number of the noun term frequencies of category j, we form a normalized weight factor for nouns in that category, in order to overcome NBC bias. This approach also enables as to properly estimate the similarity of a random document to a category, since a great number of significant words of a category in a random document (high-weighted word observation), denotes greater similarity between the document and the category. The similarity factor of our approach is based on the probability of a random document D to belong in category j, if word i is present in document D. Given that word i is assigned a weight per class j, then this probability is calculated by

$$P(D \in j \mid i \in D) = \frac{w_{i,j}}{\sum_{j=0}^{n} w_{i,j}} \tag{8}$$

where n is the total number of categories the system can identify. This metric takes into account the

cross-class importance of the words.

For example, let a two class system (category A and B) comprised of n words occurring m times, and word C appearing in both categories once. The weight of word C in both categories is 1/m. The probability that given word C a random document to belong category A (which is equal in this example to the document belonging to category B) is according to the last formula:

$$P(D \in A \mid C \in D) = \frac{w_{C,A}}{w_{C,A} + w_{C,B}} = \frac{\frac{1}{m}}{\frac{1}{m} + \frac{1}{m}} = \frac{1}{2} \tag{9}$$

which was expected as both A and B were equal in size classes and word A only occurred once in both of the them.

The evaluation criterion that denotes a document into a category is called similarity factor (*sf*) and is calculated by

$$sf(D, j) = \frac{\sum_{i=0}^{Dwords} \frac{w_{i,j}}{\sum_{j=0}^{m} w_{i,j}}}{Dwords} \tag{10}$$

where *m* is the total number of categories available and *Dwords* the words of document *D*. Since *sf* is computed on the observed set of nouns extracted by the document, and not all words appear in category j, the contribution for each word present in the document is either 0 if the word is not present in category j or calculated according to (8). Dividing by *Dwords* gives us the Expected Value for each category j. The system decides that the document belongs to the category which maximizes the similarity factor *sf*

$$D \in j \Leftarrow argmax(sf(D, j)) \tag{11}$$

In the next section we provide our methodology in pseudo-code and analyze each step of the classification process.

4. METHODOLOGY

We have developed a system based on the aforementioned approach. The system is based on a learning step and a test set, while information undergoes a preparatory suffix stripping phase. Initially every document undergoes a stemming procedure. This produces sets of stems of words that are provided as input to the system. The stemming procedure is also responsible for isolating nouns from other parts of speech. The second step of the system is the education phase where the system is provided with the example set that is going to be used to evaluate word significance per available class. The third step is the classification phase, where the system takes as input a random document and classifies it into one or more of the available categories. All systems were developed using Java ™ technology, and will become widely available upon completion of research. The system is intended to be used as a newspaper article classifier for Greek language.

4.1 Stemming Step

Stemming is the process of identifying the stem of a word (the part that does not alter in the different forms that a word may be found in a random document) from its suffix (ending). Initial work in the area was undertaken by Porter in (1980) where he proposed a system for suffix stripping for English language based on grammatical features. This work was very important as it has been used in a number of Machine Learning applications as document classification – where Scott and Matwin in (1999) vividly state that it is almost always used- and summarization

The initial work for stemming in Greek language was undertaken by Kalamboukis (1993), who adapted Porter's work for Greek language, based on greek grammatical features. Kalamboukis work included the gathering of all potential endings of suffixes of words and the extraction of the stem of the word. His work was used as a basis for our research on the area of stemming. We decided to develop a stemmer (Mamakis, Malamos, Kaliakatsos, Axaridou, Ware, 2005) that would extend Kalamboukis approach by performing minimal part of speech tagging work. There are two main reasons for that, the first one being the fact that we only wanted noun identification and the second that now efficient Greek POS tagger exists. The second point is very crucial to our approach, since while our stemmer may include non-noun information in the data sets it never fails to correctly identify a Greek noun. All other surplus data is either ignored in the classification stage, or is already existent in the training phase and therefore has already affected the importance factor of each word. On the other hand, failing to correctly identify a noun (as in (AUEB Greek POS Tagger, 2010), although the system is trainable and therefore may be suitable for the task in a costly manner) breaks the second assumption we made in the previous section. Our stemmer may include insignificant data in the final system, mostly due to existence of Greek language particularities - common suffixes between nouns and adjectives (e.g. ναυτικός – sailor and τακτικός - tactical, the first being a noun while the latter an adjective) or nouns often used as adverbs (e.g. αλήθεια – truth as a noun and really as an adverb). Still, these particularities on a more generalized aspect do not constitute a major problem on the overall efficiency of the system. The stemming step algorithm operates as follows:

Let random document D.
Split D in words w_D
For each l in w_D
Remove if Article
Remove if Preposition
Remove if Pronoun
Remove if Adverb
Remove if Verb
Check if Noun
If yes, Remove if Participle else keep
If no remove
Gather updated table of words w_D

Figure 1. Stemming algorithm

The final table w_D stands for the document to be included in training or classification.

4.2 Training Step

The training step is responsible for building up the dictionaries of words used in classification. Each newspaper article in each category is stemmed and the according article noun-table is used to create the weighted category dictionary. Each word is assigned a normalized weight according to the number of nouns present in each category. The training module algorithm is depicted in the following

table

For each category j formed by documents D_j
Let random document D
Stem D and acquire word table w_D
For each word i in w_D
Check unique word occurrence and create vector comprised by words, occurrences
For each i in updated w_D
Check if $dict_j$ (the dictionary of j category) contains i
If contains i update occurrences of i
Else append i in $dict_j$ with provided occurrences.
If category j has no more documents create weight feature.

Figure 2. Training Module Algorithm

4.3 Classification Step

The classification step is responsible for assigning a random document in one category. Each random document is stemmed and the resulting noun table is checked in each category to form the similarity factor. The category a document belongs to is defined by the maximum similarity factor. The classification step algorithm operates as follows:

Let random document D.
Stem D and acquire word table w_D
For each category j
For each word i in w_D
Check if i is contained in $dict_j$
Add to similarity measure of document-category sf_{Dj} the word weight for this category $w_{i,j}$
If document has no more words divide sf_{Dj} by the size of w_D
D belongs to category sf_{Dj}

Figure 3. Classification Step

5. EVALUATION RESULTS

We have tested our system against two statistical algorithms, NBC and statistical Language Models, as provided by Mallet (McCallum, 2002) and Lingpipe (ALIAS-I, 2008) natural language processing toolkits. The statistical Language Models utilized included a 6-gram Language Model and a trigram Language Model.

5.1 Corpus Profiles

The training and test dataset was manually gathered from online Greek news sites and was initially classified according to the classification scheme used by each site. The articles were post-processed in order to fit in our classification scheme that was made up by six categories namely: Business and Finance, Culture, Health, Politics, Science and Technology and Sports. The training corpus was comprised by 1015 articles while the test corpus by 353 articles. Each article in the corpus underwent the stemming procedure and the resulting categories had the characteristics depicted in Table 1.

Category	# of unique words	# of total words	Average word occurrence	Average weight
Business & Finance	5686	59777	10,51	0,000176
Culture	5750	26932	4,68	0,000174
Health	1181	3073	2,60	0,000847
Politics	7059	63218	8,96	0,000142
Science & Technology	3593	24429	6,80	0,000278
Sports	4696	15412	3,28	0,000213
Totals	27965	192841	6,139254653	0,000304909

Table 1. Nouns per category

As it may be observed, each category is formed from around 3500 to 7100 unique words, except for Health class which is comprised by only 1181 unique nouns. The total number of words shows the total number of nouns included in each category (unique words times word frequency). Each category depicts its own average word occurrence and according word weight. The average word weight ($w_{i,j}$), excluding Health class, is around 0.000142 and 0.000278 depending on number of total words, while in Health class the average weight is 0.000847, denoting that a word present in Health class is 3 to 5 times more important than this word on any other domain.

5.2 Experiments

All classifiers were provided as input stems of words in the training step and stems of nouns in the sentences, in exactly the same manner. Thus, a number of interesting results were acquired. The complete results showed that our algorithm outperformed both Naive Bayes Classifier, 6-gram and trigram Language Model as it is shown in Table 2.

		Our Classifier	NBC	LM-6	LM-3
Positive	#	326	302	284	294
	%	92,35	85,55	80,45	83,29
Negative	#	27	51	69	59
	%	7,65	14,45	19,55	16,71
Totals	#	353	353	353	353
	%	100	100	100	100

Table 2. Overall Classification Results

In Table 2, positive results are considered to be the ones where the systems managed to correctly match the human assigned class, whereas negative results are considered the ones that the systems failed to correctly identify. Thus, as it may be seen our algorithm outperformed both Naive Bayes Classifier and Language Models. More specifically, our classifier achieved a percentage of 92.35% correctly identified articles, while all other algorithms achieved well below 90%.

5.3 Results per Category

In the following table (Table 3), we will try to project per category efficiency of our system when compared to NBC and LMs, for single-labeled documents, as some interesting results may be extracted.

CATEGORIES		Our Classifier		NBC		LM-6		LM-3	
		#	%	#	%	#	%	#	%
Business & Finance	Positive	66	95,65	63	91,30	59	85,51	60	86,96
	Negative	3	4,35	6	8,70	10	14,49	9	13,04
	Totals	69	100,00	69	100,00	69	100,00	69	100,00
Culture	Positive	54	96,43	54	96,43	53	94,64	52	92,86
	Negative	2	3,57	2	3,57	3	5,36	4	7,14
	Totals	56	100,00	56	100,00	56	100,00	56	100,00
Health	Positive	27	62,79	10	23,26	5	11,63	13	30,23
	Negative	16	37,21	33	76,74	38	88,37	30	69,77
	Totals	43	100,00	43	100,00	43	100,00	43	100,00
Politics	Positive	47	97,92	47	97,92	46	95,83	42	87,50
	Negative	1	2,08	1	2,08	2	4,17	6	12,50
	Totals	48	100,00	48	100,00	48	100,00	48	100,00
Science & Technology	Positive	73	93,59	70	89,74	62	79,49	69	88,46
	Negative	5	6,41	8	10,26	16	20,51	9	11,54
	Totals	78	100,00	78	100,00	78	100,00	78	100,00
Sports	Positive	59	100,00	58	98,31	59	100,00	58	98,31
	Negative	0	0,00	1	1,69	0	0,00	1	1,69
	Totals	59	100,00	59	100,00	59	100,00	59	100,00

Table 3. Classification Results per Category

First of all, it is obvious that our proposed algorithm produced for the given test corpus better (or as good results in some cases) as NBC an LM. The results between the algorithms are comparable for all categories, except for Health class. Health class is made up by the smallest dictionary of all six categories, while it contains a number of words similar to Science & Technology class. As it is observed, our algorithm correctly classifies 27 out of 43 Health articles in the test corpus (almost 63%), as opposed to NBC and LM which face serious problems (23%, 12% and 30% of the total Health articles correctly classified). A reason for that is that while NBC is indifferent to word co-occurrence it also treats every class as independent to one another trying to maximize the highest scoring of them. This is the characteristic that tends to create bias towards a class with larger datasets as observed in (Rish, 2001). Health class in our test was comprised of the smallest dataset, sharing common words with Science and Technology, the latter being more than twice as big as Health corpus. Therefore, NBC tended to classify these documents incorrectly. Contrary to that, our initial target in the system was to treat each domain as equally probable, eliminating any bias. This was achieved through the weight calculation formula used to estimate word contribution to a category, through the observed frequency. In addition to that, we consider that a word may exist in more than one categories, with different weights per category. Therefore, it is important to estimate the overall importance of this word not only on one category but also on a cross-category level. These two estimates tend to produce less biased results on small datasets that share common words with larger datasets, as Health class.

5.4 Tests with Ambiguous Data

During our tests, we also observed that our algorithm produced category results that in some cases were ambiguous, especially for articles that their manual classification by the newspapers was ambiguous, since their content was semantically shared between 2 or 3 thematic areas. For example, in our test article a83 originally classified in Politics by the newspaper, dealt with the cultural effects of the elections on a country through history. This implied that while Culture is the primary class for that article, Politics could also be a potential category. In fact our classifier, identified both categories with Culture having an *sf* value of 0.2263 and Business and Finance having an *sf*=0.2214 while other

classes' *sf* were in the region of 0.0067 to 0.1469. Motivated by that, we tried to verify how the system would operate with ambiguous input data, in order to check its robustness. We gathered 23 newspaper articles that could not be classified into one category. We assigned two categories per article in order of similarity (e.g. Business and Finance/Politics denoting that this article fits into Business and Finance primarily and Politics secondarily) and run the experiment for these articles.

The results acquired are shown in Table 4.

		Our Classifier
Positive	#	21
	%	91,30
Negative	#	0
	%	0
Ambiguous	#	2
	%	8,70
Totals	#	23
	%	100

Table 4. Classification with ambiguous input data

In this case, an extra selection from positive and negative is used, namely ambiguous. In this case Positive stands for correct classification of the document in all classes in order of classification similarity, negative failure to categorize the document in any of the classes it belongs to and ambiguous either successfully categorizing into one of the categories of the document, but not all of them, or successfully categorizing into all categories but with different order from the one supplied by the human classifier. As we observe, the efficiency of the classifier correctly identifying the classes of a document is similar to the one of single-label classification. Therefore, this hints that the system could be used for multi-label classification as well, since the results are very promising. However, this is beyond the scope of current work, as this paper exclusively deals with single-label classification.

6. CONCLUSIONS

In this paper, we presented a statistical approach for newspaper article classification that significantly outperforms both NBC and LMs. This approach is based on the same fundamentals as NBC, yet instead of utilizing the product of each NBC feature, our approach uses the sum of each feature probability. This reduces error propagation and bias towards specific categories, constituted by large datasets. This approach, also, enables both single-label and multi-label classification, as the algorithm engages a normalized similarity factor, that empirically was found to approximate effectively cross-class classification. The proposed algorithm is part of a text summarization system for Greek language. Another important remark from our experiments regarding classification was that Language Models efficiency increased when setting the n-gram size to three from six. Potentially their performance will increase if we use a bigram, but still it is not expected to outperform our approach. Future work includes the evaluation of the system on multi-label classification tasks, driven by the fact that example results on a very small corpus supplied promising hints on the overall performance of the classifier. Yet, this is beyond the scope of this paper, and the algorithm has to be evaluated with statistical multi-label algorithms.

REFERENCES

ALIAS-I, (2008), *LingPipe 4.0.0*. http://alias-i.com/lingpipe

AUEB Greek POS Tagger, (2010),

http://nlp.cs.aueb.gr/software_and_datasets/AUEB_Greek_POS_tagger.tar.gz, last accessed 20/7/2010

BOURAS C., and TSOGKAS V., (2010), "Improving Text Summarization Using Noun Retrieval Techniques", *in Lecture Notes in Computer Science, Volume 5178/2010, pp 593-600,* Springer Berlin / Heidelberg, ISSN: 1611-3349

CLARK P., and NIBBLET T., (1989), "The CN2 Induction Algorithm", in *Machine Learning, Vol 3(4), pp 261-283*, Springer Netherlands, ISSN: 1573-0565

CROFT W. B., (2003), "Language models for Information Retrieval", in *Proceedings of the 19th International Conference on Data Engineering*, Bangalore, India

DAI W., XUE G. R., YANG Q., and YU Y., (2007), "Transferring Naive Bayes Classifiers for Text Classification", in *Proceedings of the 22nd AAAI Conference on Artificial Intelligence, pp 540-545,* AAAI Press, Vancouver, Canada

DUCHROW T., SCHTATLAND T., GUETTLER D., PIVOVAROV M., KRAMER S., and WEISSLEDER R., (2009), *"Enhancing Navigation in biomedical databases by community driven voting and database-driven text classification"*, in *BMC Bioinformatics 2009, Vol 10, pp317*

GALLEY M., MCKEOWN K., (2003), "Improving Word Sense Disambiguation in Lexical Chaining", *in Proceedings of the 18th international joint conference on Artificial Intelligence, pp 1486-1488,* Acapulco, Mexico

HAN E. H., KARYPIS G., and KUMAR V., (2001), "Text categorization using weight adjusted k-nearest neighbor classification", in *Advances in Knowledge Discovery and Data Mining, LNCS Vol 2035, pp 53-65*, Springer Verlag, ISSN: 0302-9743

JOACHIMS T., (1997), "A probabilistic analysis of the Rocchio Algorithm with Tfidf for text categorization", in *Proceedings of the 14th International Conference on Machine Learning, pp 143-151*,San Francisco, USA

KALAMBOUKIS T.Z., (1993), "Suffix stripping with modern greek", in *Program: electronic library and information systems, Vol. 29(3), pp.313 - 321*, ISSN:0033-0337

KOTSIANTIS S. B., and PINTELAS P. E. (2004), "Increasing the Classification Accuracy of Simple Bayesian Classifier", *Lecture Notes in Artificial Intelligence, AIMSA 2004*, Springer-Verlag Vol 3192, pp.198–207

KWON O. W., and LEE J. H., (2003),"Text Categorization based on k-nearest neighbor approach for website classification", in *Information Processing and Management, Volume 39, Issue 1, pp25-44,* Elsevier

MAMAKIS G., MALAMOS A.G., KALIAKATSOS Y., AXARIDOU A., and WARE A., (2005), "An algorithm for automatic content summarization in modern greek language", in *Proceedings of ICICT '05,* Cairo, Egypt

MCCALLUM A. K., and NIGAM K. (1998), "A comparison of Event Models for Naive Bayes Text Classification", in *Proceedings of AAAI-98 Workshop on Learning for Text Categorization, pp 41-48,* AAAI Press, Winsconsin, USA

MCCALLUM A. K., (2002), "MALLET: A Machine Learning for Language Toolkit.", http://mallet.cs.umass.edu.

MIAO D., DUAN Q., ZHANG H., and JIAO N., (2009), "Rough set based hybrid algorithm for text classification", in *Expert Systems with Applications, Volume 36, Issue 5, pp 9168-9174*, Elsevier

MOSCHITTI A., (2003), "A study on Optimal Parameter for Rocchio Text Classifier", in *Advances in Information Retrieval, LNCS Vol 2633, pp 546-547* , Springer/Heidelberg, ISSN:1611-3349

NIGAM K., MCCALLUM A. K., THRUN S., and MITCHELL T. (2000), "Text Classification from Labeled and Unlabeled Documents using EM", in *Machine Learning, Vol 23, Issue 2-3, Special Issue on Information Retrieval, pp 103-134*, Kluwer Academic Publishers, ISSN:0885-6125

PENG F., SCHUURMANS D., and WANG S., (2004), "Augmenting Naive Bayes Classifiers with Statistical Language Models", in *Information Retrieval, Vol 7, Issues 3-4, pp 317-345,* Kluwer

Academic Publishers, ISSN:1573-7659
PONTE J. M., and CROFT W. B., (1998), "A language modelling approach to information retrieval", in *Proceedings of the 21st annual international ACM SIGIR conference on Research and Development in Information Retrieval, pp 275-281*, Melbourne, Australia
PORTER M. F., (1980), "An algorithm for suffix stripping", *in Program, Vol14(3), pp 130-137*
RENNIE J. D. M., SHIH L., TEEVAN J., and KARGER D. R. (2003), "Tackling the poor assumptions of Naive Bayes Text Classifiers", in *Proceedings of the 20th International Conference on Machine Learning (ICML-2003)*, Washington D.C., USA
REUTERS CORPUS, (2010), http://trec.nist.gov/data/reuters/reuters.html, last accessed 30/7/2010
RISH I. (2001), "An empirical study of the Naive Bayes Classifier", in *Proceedings of IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence*, pp 41-46, Seattle, USA
ROCCHIO J., (1971), "Relevance Feedback in Information Retrieval", in *the SMART Retrieval System, Experiments in Automatic Document Processing, pp 313-323*
RUGGIERI S., (2002), "Efficient C4.5", in *IEEE Transactions on Knowledge and Data Enginnering, Vol 14(2), pp 438-444*
RUIZ M. E., and SRINIVASAN P., (2002), "Hierarchical text categorization using neural networks", *Information Retrieval, Vol 5, pp 87-118*, Springer
SCOTT S., and MATWIN S., (1999), "Feature Engineering for Text Classification", in *Proceedings of the Sixteenth International Conference on Machine Learning, pp 379-388*, Morgan Kauffman Publishers, ISBN:1-55860-612-2
SRIKANTH M., and SRIHARI R., (2002), "Biterm language models for document retrieval", in *Proceedings of the 25th annual international ACM SIGIR conference on Research and Development in Information Retrieval, pp 425-426*, Tampere, Finland
TAN S., (2005),"Neighbor-weighted k-nearest neighbor for unbalanced text corpus", in *Expert System with application, Vol 28, Issue 4, pp 667-671*, Elsevier
TREC CORPUS, (2010), http://trec.nist.gov, last accessed 30/7/2010
TSOUMAKAS G., KATAKIS I., and VLAHAVAS I. (2006), "A review of multilabel classification methods", in *Proceedings of the 2nd ADBIS Workshop on Data Mining and Knowledge Discovery*, pp. 99-109.
ZHANG M. L., and ZHU Z. H., (2006), "Multilabel Neural Networks with applications to functional genomics and Text categorization", *in IEEE Transactions on Knowledge and Data Engineering, Vol 18 (10), pp 1338-1351*
ZHU J., WANG H., YAO T., and TSOU B. K., (2008),"Active Learning with sampling by uncertainty and density for word sense disambiguation and text classification", in *Proceedings of the 22nd International Conference on Computational Linguistics, Volume 1, pp1137-1144*, Manchester, UK