Text Classification: A Recent Overview

M. IKONOMAKIS
Department of Mathematics
University of Patras, GREECE

S. KOTSIANTIS
Department of Mathematics
University of Patras, GREECE

V. TAMPAKAS
Technological Educational
Institute of Patras, GREECE

Abstract: Text classification is becoming more and more important with the rapid growth of on-line information available. This paper describes the text classification process. Of course, a single article cannot be a complete review of the text classification domain. Despite this, we hope that the references cited cover the major theoretical issues and guide the researcher to interesting research directions.

Key-Words: text mining, learning algorithms, feature selection, text representation

1 Introduction

Intuitively Text Classification is the task of classifying a document under a predefined category. More formally, if d_i is a document of the entire set of documents D and $\{c_1, c_2, ..., c_n\}$ is the set of all the categories, then text classification assigns one category c_j to a document d_i.

As in every supervised machine learning task, an initial dataset is needed. A document may be assigned to more than one category (Ranking Classification), but in this paper only researches on Hard Categorization (assigning a single category to each document) are taken into consideration. Moreover, approaches, that take into consideration other information besides the pure text, such as hierarchical structure of the texts or date of publication, are not presented. This is because the main issue of this paper is to present techniques that exploit the most of the text of each document and perform best under this condition, since a hierarchical structure of the documents.

Sebastiani gave an excellent review of text classification domain [22]. Thus, in this work apart from the brief description of the text classification we refer to some more recent works than those in Sebastiani's article as well as few articles that were not referred by Sebastiani. In Figure 1 is given the graphical representation of the Text Classification process.

The task of constructing a classifier for documents does not differ a lot from other tasks of Machine Learning. The main issue is the representation of a document [14]. In Section 2 the document representation is presented. One particularity of the text categorization problem is that the number of features (unique words or phrases) can easily reach orders of tens of thousands. This raises big hurdles in applying many sophisticated learning algorithms to the text categorization

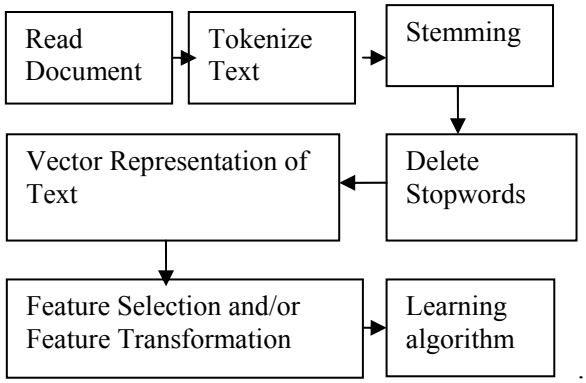

Fig. 1. Text Classification Process

Thus dimension reduction methods are called for. Two possibilities exist, either selecting a subset of the original features [3], or transforming the features into new ones, that is, computing new features as some functions of the old ones [9]. We examine both in turn in Section 3 and Section 4. After the previous steps a Machine Learning algorithm can be applied. Some algorithms have been proven to perform better in Text Classification tasks and are more often used; such as Support Vector Machines. A brief description of recent modification of learning algorithms in order to be applied in Text Classification is given in Section 5. There are a number of methods to evaluate the performance of a machine learning algorithms in Text Classification. Most of these methods are described in Section 6. Some open problems are mentioned in the last section.

2 Vector space document representations

A document is a sequence of words [14]. So each document is usually represented by an array of words. The set of all the words of a training set is called vocabulary, or feature set. So a document

*The Project is co-funded by the European Social Fund and National Resources.

can be presented by a binary vector, assigning the value 1 if the document contains the feature-word or 0 if the word does not appear in the document. This can be translated as positioning a document in a $R^{|V|}$ space, were $|V|$ denotes the size of the vocabulary V .

Not all of the words presented in a document can be used in order to train the classifier [17]. There are useless words such as auxiliary verbs, conjunctions and articles. These words are called stopwords. There exist many lists of such words which are removed as a preprocess task. This is done because these words appear in most of the documents.

Stemming is another common preprocessing step. In order to reduce the size of the initial feature set is to remove misspelled or words with the same stem. A stemmer (an algorithm which performs stemming), removes words with the same stem and keeps the stem or the most common of them as feature. For example, the words “train”, “training”, “trainer” and “trains” can be replaced with “train”. Although stemming is considered by the Text Classification community to amplify the classifiers performance, there are some doubts on the actual importance of aggressive stemming, such as performed by the Porter Stemmer [22].

An ancillary feature engineering choice is the representation of the feature value [14]. Often a Boolean indicator of whether the word occurred in the document is sufficient. Other possibilities include the count of the number of times the word occurred in the document, the frequency of its occurrence normalized by the length of the document, the count normalized by the inverse document frequency of the word. In situations where the document length varies widely, it may be important to normalize the counts. Further, in short documents words are unlikely to repeat, making Boolean word indicators nearly as informative as counts. This yields a great savings in training resources and in the search space of the induction algorithm. It may otherwise try to discretize each feature optimally, searching over the number of bins and each bin’s threshold.

Most of the text categorization algorithms in the literature represent documents as collections of words. An alternative which has not been sufficiently explored is the use of word meanings, also known as senses. Kehagias et al. using several algorithms, they compared the categorization accuracy of classifiers based on words to that of classifiers based on senses [11]. The document collection on which this comparison took place is a subset of the annotated Brown Corpus semantic concordance. A series of experiments indicated that the use of senses does not result in any significant categorization improvement.

3 Feature Selection

The aim of feature-selection methods is the reduction of the dimensionality of the dataset by removing features that are considered irrelevant for the classification [5]. This transformation procedure has been shown to present a number of advantages, including smaller dataset size, smaller computational requirements for the text categorization algorithms (especially those that do not scale well with the feature set size) and considerable shrinking of the search space. The goal is the reduction of the curse of dimensionality to yield improved classification accuracy. Another benefit of feature selection is its tendency to reduce overfitting, i.e. the phenomenon by which a classifier is tuned also to the contingent characteristics of the training data rather than the constitutive characteristics of the categories, and therefore, to increase generalization.

Methods for feature subset selection for text document classification task use an evaluation function that is applied to a single word [24]. All words are independently evaluated and sorted according to the assigned criterion. A predefined number of the best features are taken to form the best feature subset. Scoring of individual words can be performed using some of the measures, for instance, document frequency, term frequency, mutual information, information gain, odds ratio, χ^2 statistic and term strength [3], [26], [5], [25], [24]. In Table of the most common metrics are presented together with new metrics. Many of the common used metrics have their origin in Text Retrieval or Machine Learning. The newly presented metrics are often modification of common metrics.

As we have already mentioned Best Individual Features (BIF) methods evaluate all the n words individually according to a given criterion, sort them and select the best k words. Sequential forward selection (SFS) methods firstly select the best single word evaluated by given criterion [18]. Then, add one word at a time until the number of selected words reaches desired k words. However SFS methods do not result in the optimal words subset but they take note of dependencies between words as opposed to the BIF methods. Therefore SFS often give better results than BIF. However, SFS are not usually used in text classification

because of their computation cost due to large vocabulary size.

Forman has present benchmark comparison of 12 metrics on well known training sets [5].

Metrics mathematical forms	
Information Gain	$IG(t) = -\sum_{i=1}^{m} P(c_i)\log P(c_i) + P(t)\sum_{i=1}^{m} P(c_i \mid t)\log P(c_i \mid t) + P(\bar{t})\sum_{i=1}^{m} P(c_i \mid \bar{t})\log P(c_i \mid \bar{t})$
Gain Ratio	$GR(t_k, c_i) = \frac{\sum_{c\in\{c_i,\bar{c}_i\}}\sum_{t\in\{t_k,\bar{t}_k\}} P(t,c)\log\frac{P(t,c)}{P(t)P(c)}}{-\sum_{c\in\{c_i,\bar{c}_i\}} P(c)\log P(c)}$
Term Frequency	$tf(f_i, d_i) = \frac{freq_{ij}}{\max_k freq_{kj}}$
Inversed Document Frequency	$idf_i = \log\frac{\lvert document\_set\rvert}{\lvert documents\_were\_f_i\_appears\rvert}$
Chi-square	$Chi(f_i, c_j) \equiv \chi^2(f_i, c_j) = \frac{N\times(AD-CB)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)}$
Term Strength	$s(t) = P(t \in y \mid t \in x)$
Weighted Ratio	$WOddsRatio(w) = P(w)\times OddsRatio(w)$
OddsRatio	$OddsRatio(f_i, c_j) = \log\frac{P(f_i \mid c_j)(1-P(f_i \mid \neg c_j))}{(1-P(f_i \mid c_j))P(f_i \mid \neg c_j)}$
Logarithmic Probability Ratio	$LogProbRatio(w) = \log\frac{P(w \mid c)}{P(w \mid \neg c)}$
Pointwise Mutual Information	$I(x, y) = \log\frac{p(x,y)}{p(x)p(y)}$
Odds Numerator	$OddsNum(w, c) = P(w \mid c)(1-P(w \mid \neg c))$
Probability Ratio	$\Pr R(w \mid c) = \frac{P(w \mid c)}{P(w \mid \neg c)}$
Bi-Normal Separation	$F^{-1}(P(w \mid c)) - F^{-1}(P(w \mid \neg c))$
Pow	$(1-P(w \mid \neg c))^k - (1-P(w \mid c))^k$
Weight of evidence for Text	$Weight(w) = \sum_i P(c_i)\times P(w)\times\left\lvert\log\frac{P(c_i \mid w)(1-P(c_i))}{P(c_i)(1-P(c_i \mid w))}\right\rvert$

Table 1. Feature Selection metrics

According to Forman, BNS performed best by wide margin using 500 to 1000 features, while Information Gain outperforms the other metrics when the features vary between 20 and 50. Accuracy 2 performed equally well as Information Gain. Concerning the performance of chi-square, it was consistently worse the Information Gain. Since there is no metric that performs constantly better than all others, researchers often combine two metrics in order to benefit from both metrics [5].

Although machine learning based text classification is a good method as far as

performance is concerned, it is inefficient for it to handle the very large training corpus. Thus, apart from feature selection, many times instance selection is needed. Guan and Zhou proposed a training-corpus pruning based approach to speedup the process [7]. By using this approach, the size of training corpus can be reduced significantly while classification performance can be kept at a level close to that of without training documents pruning according to their experiments. Fragoudis et al. integrated Feature and Instance Selection for Text Classification [6].

4 Feature Transformation

Feature Transformation varies significantly from Feature Selection approaches, but like them its purpose is to reduce the feature set size [9]. This approach does not weight terms in order to discard the lower weighted but compacts the vocabulary based on feature concurrencies. Latent Semantic Indexing (LSI) infers the dependence among the original features based on the dataset and passes this dependence on to a newly obtained feature set [20]. This approach is not intuitive discernible for a human but has a good performance.

Principal Component Analysis is a similar method [29]. Its aim is to learn a discriminative transformation matrix in order to reduce the initial feature space into a lower dimensional feature space in order to reduce the complexity of the classification task without any trade-off in accuracy. The transform is derived from the eigenvectors corresponding. The covariance matrix of data in PCA corresponds to the document term matrix multiplied by its transpose. Entries in the covariance matrix represent co-occurring terms in the documents. Eigenvectors of this matrix corresponding to the dominant eigenvalues are now directions related to dominant combinations can be called “topics” or “semantic concepts”. A transform matrix constructed from these eigenvectors projects a document onto these “latent semantic concepts”, and the new low dimensional representation consists of the magnitudes of these projections. The eigenanalysis can be computed efficiently by a sparse variant of singular value decomposition of the document-term matrix [10].

5 Machine learning algorithms

After feature selection and transformation the documents can be easily represented in a form that can be used by a ML algorithm. Such algorithms often used in Text Classification are Naive Bayes, SVM, and kNN and decision trees.

Naive Bayes is often used in text classification applications and experiments because of its simplicity and effectiveness. However, its performance is often degraded because it does not model text well. Schneider addressed the problems and show that they can be solved by some simple corrections [21]. Klopotek and Woch presented results of empirical evaluation of a Bayesian multinet classifier based on a new method of learning very large tree-like Bayesian networks [13]. The study suggests that tree-like Bayesian networks are able to handle a text classification task in one hundred thousand variables with sufficient speed and accuracy.

Support vector machines (SVM), when applied to text classification provide excellent precision, but poor recall. One means of customizing SVMs to improve recall, is to adjust the threshold associated with an SVM. Shanahan and Roma described an automatic process for adjusting the thresholds of generic SVM [23].

Johnson et al. described a fast decision tree construction algorithm that takes advantage of the sparsity of text data, and a rule simplification method that converts the decision tree into a logically equivalent rule set [8].

Lim proposed a method which improves performance of kNN based text classification by using well estimated parameters [16]. Some variants of the kNN method with different decision functions, k values, and feature sets were proposed and evaluated to find out adequate parameters.

When training a binary classifier per category in text categorization, we use all the documents in the training corpus that belong to that category as relevant training data and all the documents in the training corpus that belong to all the other categories as non-relevant training data. It is often the case that there is an overwhelming number of non relevant training documents especially when there is a large collection of categories with each assigned to a small number of documents, which is typically an “imbalanced data problem". This problem presents a particular challenge to classification algorithms, which can achieve high accuracy by simply classifying every example as negative. To overcome this problem, cost sensitive learning is needed [4].

A scalability analysis of a number of classifiers in text categorization is given in [28]. Vinciarelli presents categorization experiments performed over noisy texts [27]. By noisy it is meant any text obtained through an extraction process (affected by

errors) from media other than digital texts (e.g. transcriptions of speech recordings extracted with a recognition system). The performance of the categorization system over the clean and noisy (Word Error Rate between ~10 and ~50 percent) versions of the same documents is compared. The noisy texts are obtained through Handwriting Recognition and simulation of Optical Character Recognition. The results show that the performance loss is acceptable.

In the context of combining multiple classifiers for text categorization, a number of researchers have shown that combining different classifiers can improve classification accuracy [1], [2]. Nardiello et al. proposed algorithms in the family of "boosting"-based learners for automated text classification [19].

6 Evaluation

There are various methods to determine effectiveness; however, precision, recall, and accuracy are most often used. To determine these, one must first begin by understanding if the classification of a document was a true positive (TP), false positive (FP), true negative (TN), or false negative (FN) (see Table 2).

TP	Determined as a document being classified correctly as relating to a category.
FP	Determined as a document that is said to be related to the category incorrectly.
FN	Determined as a document that is not marked as related to a category but should be.
TN	Documents that should not be marked as being in a particular category and are not.

Table 2. Classification of a document

Precision (π_i) is determined as the conditional probability that a random document d is classified under c_i, or what would be deemed the correct category. It represents the classifiers ability to place a document as being under the correct category as opposed to all documents place in that category, both correct and incorrect: $\pi_i = \frac{TP_i}{TP_i+FP_i}$

Recall (ρ_i) is defined as the probability that, if a random document d_x should be classified under category (c_i), this decision is taken.

$$\rho_i = \frac{TP_i}{TP_i+FN_i}$$

Accuracy is commonly used as a measure for categorization techniques. Accuracy values, however, are much less reluctant to variations in the number of correct decisions than precision and recall:

$$A_i = \frac{TP_i+TN_i}{TP_i+TN_i+FP_i+FN_i}$$

Many times there are very few instances of the interesting category in text categorization. This overrepresentation of the negative class in information retrieval problems can cause problems in evaluating classifiers' performances using accuracy. Since accuracy is not a good metric for skewed datasets, the classification performance of algorithms in this case is measured by precision and recall [4].

Reuters Corpus Volume I (RCV1) is an archive of over 800,000 manually categorized newswire stories recently made available by Reuters, Ltd. for research purposes [15]. Using this collection, we can compare the learning algorithms.

7 Conclusion

It has observed that even for a specified classification method, classification performances of the classifiers based on different training text corpuses are different; and in some cases such differences are quite substantial. This observation implies that a) classifier performance is relevant to its training corpus in some degree, and b) good or high quality training corpuses may derive classifiers of good performance. Unfortunately, up to now little research work in the literature has been seen on how to exploit training text corpuses to improve classifier's performance.

Moreover, there are other two open problems in text mining: polysemy, synonymy. Polysemy refers to the fact that a word can have multiple meanings. Distinguishing between different meanings of a word (called word sense disambiguation) is not easy, often requiring the context in which the word appears. Synonymy means that different words can have the same or similar meaning.

References:

[1] Bao Y. and Ishii N., "Combining Multiple kNN Classifiers for Text Categorization by Reducts", LNCS 2534, 2002, pp. 340-347

[2] Bi Y., Bell D., Wang H., Guo G., Greer K., "Combining Multiple Classifiers Using Dempster's Rule of Combination for Text Categorization", MDAI, 2004, 127-138.

[3] Brank J., Grobelnik M., Milic-Frayling N., Mladenic D., "Interaction of Feature Selection Methods and Linear Classification Models",

Proc. of the 19th International Conference on Machine Learning, Australia, 2002

[4] Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P., "SMOTE: Synthetic Minority Over-sampling Technique," Journal of AI Research, 16 2002, pp. 321-357.

[5] Forman, G., An Experimental Study of Feature Selection Metrics for Text Categorization. Journal of Machine Learning Research, 3 2003, pp. 1289-1305

[6] Fragoudis D., Meretakis D., Likothanassis S., "Integrating Feature and Instance Selection for Text Classification", SIGKDD '02, July 23-26, 2002, Edmonton, Alberta, Canada.

[7] Guan J., Zhou S., "Pruning Training Corpus to Speedup Text Classification", DEXA 2002, pp. 831-840

[8] D. E. Johnson, F. J. Oles, T. Zhang, T. Goetz, "A decision-tree-based symbolic rule induction system for text categorization", IBM Systems Journal, September 2002.

[9] Han X., Zu G., Ohyama W., Wakabayashi T., Kimura F., Accuracy Improvement of Automatic Text Classification Based on Feature Transformation and Multi-classifier Combination, LNCS, Volume 3309, Jan 2004, pp. 463-468

[10] Ke H., Shaoping M., "Text categorization based on Concept indexing and principal component analysis", Proc. TENCON 2002 Conference on Computers, Communications, Control and Power Engineering, 2002, pp. 51-56.

[11] Kehagias A., Petridis V., Kaburlasos V., Fragkou P., "A Comparison of Word- and Sense-Based Text Categorization Using Several Classification Algorithms", JIIS, Volume 21, Issue 3, 2003, pp. 227-247.

[12] Kim S. B., Rim H. C., Yook D. S. and Lim H. S., "Effective Methods for Improving Naive Bayes Text Classifiers", LNAI 2417, 2002, pp. 414-423

[13] Klopotek M. and Woch M., "Very Large Bayesian Networks in Text Classification", ICCS 2003, LNCS 2657, 2003, pp. 397-406

[14] Leopold, Edda & Kindermann, Jörg, "Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?", Machine Learning 46, 2002, pp. 423 - 444.

[15] Lewis D., Yang Y., Rose T., Li F., "RCV1: A New Benchmark Collection for Text Categorization Research", Journal of Machine Learning Research 5, 2004, pp. 361-397.

[16] Heui Lim, Improving kNN Based Text Classification with Well Estimated Parameters, LNCS, Vol. 3316, Oct 2004, Pages 516 - 523.

[17] Madsen R. E., Sigurdsson S., Hansen L. K. and Lansen J., "Pruning the Vocabulary for Better Context Recognition", 7th International Conference on Pattern Recognition, 2004

[18] Montanes E., Quevedo J. R. and Diaz I., "A Wrapper Approach with Support Vector Machines for Text Categorization", LNCS 2686, 2003, pp. 230-237

[19] Nardiello P., Sebastiani F., Sperduti A., "Discretizing Continuous Attributes in AdaBoost for Text Categorization", LNCS, Volume 2633, Jan 2003, pp. 320-334

[20] Qiang W., XiaoLong W., Yi G., "A Study of Semi-discrete Matrix Decomposition for LSI in Automated Text Categorization", LNCS, Volume 3248, Jan 2005, pp. 606-615.

[21] Schneider, K., Techniques for Improving the Performance of Naive Bayes for Text Classification, LNCS, Vol. 3406, 2005, 682-693.

[22] Sebastiani F., "Machine Learning in Automated Text Categorization", ACM Computing Surveys, vol. 34 (1),2002, pp. 1-47.

[23] Shanahan J. and Roma N., Improving SVM Text Classification Performance through Threshold Adjustment, LNAI 2837, 2003, 361-372

[24] Soucy P. and Mineau G., "Feature Selection Strategies for Text Categorization", AI 2003, LNAI 2671, 2003, pp. 505-509

[25] Sousa P., Pimentao J. P., Santos B. R. and Moura-Pires F., "Feature Selection Algorithms to Improve Documents Classification Performance", LNAI 2663, 2003, pp. 288-296

[26] Torkkola K., "Discriminative Features for Text Document Classification", Proc. International Conference on Pattern Recognition, Canada, 2002.

[27] Vinciarelli A., "Noisy Text Categorization, Pattern Recognition", 17th International Conference on (ICPR'04) , 2004, pp. 554-557

[28] Y. Yang, J. Zhang and B. Kisiel., "A scalability analysis of classifiers in text categorization", ACM SIGIR'03, 2003, pp 96-103

[29] Zu G., Ohyama W., Wakabayashi T., Kimura F., "Accuracy improvement of automatic text classification based on feature transformation": Proc: the 2003 ACM Symposium on Document Engineering, November 20-22, 2003, pp.118-120