Automatic Categorization of Fanatic Texts

Jiří Kléma KLEMA@LABE.FELK.CVUT.CZ
Department of Cybernetics, CTU Prague, Technická 2, 166 27 Prague, Czech Republic

Ahmad Almonayyes SAMI@MCS.SCI.KUNIV.EDU.KW
Dept. of Mathematics & Computer Science, Kuwait University, P.O.Box: 5969, Safat, Kuwait 13060

Abstract

This paper presents a task of automatic categorization of fanatic texts. The analyzed set of texts stems from an Arabic environment in Kuwait, where teachers and students were asked questions regarding various terrorist tendencies. The responses were classified by a domain expert into one of three classes with respect to degree of fanaticism of their content. The main task was to develop an automated tool, which is able to grasp the implicit expert's knowledge and distinguish the documents according to their content, i.e., a classifier. The paper deals with the bag-of-words representation of the documents. It applies learning algorithms that proved to work well in the field of text categorization (TFIDF classifier, multinomial probabilistic model) as well as the random forest classifier that is well-known to cope with domains described by a large number of features. The associated task was to discover any knowledge helping to understand the domain. For this reason, the final models were also analyzed and used to reveal inherent structure inside the set of documents (a sub-class structure) or to identify important words and their possible relations.

1. Introduction

The knowledge of foreign language plays a big role in intelligence and counter-terrorism. The intelligence community relies heavily on language to create finished intelligence products for decision makers. The information is gathered from intelligence reports, embassy reporting, media news, internet which is now increasingly in non-English languages, or other resources. Of course, the finished product is in English, but the input may come from several different foreign languages and need to be evaluated by a range of people with the ability to translate and interpret the data in its original language within its particular context. A lack of language skills can limit intelligence analyst insight into foreign culture, constraining their ability to understand and anticipate deterioration in a particular situation, and hence, endangering national security readiness to confront a potential danger. For example, it has become clearer than ever that those events in the Middle East affect our daily lives. The world today faces a critical shortage of linguistically competent professionals to assist intelligence analysts in classifying Arabic-written documents (e.g. emails) which may contain information that would be harmful to the world stability. Arabic is considered a difficult language to learn due to the fact that it has many forms, the modern standard (the written language), and Arabic dialect (the spoken form in one country or region). Therefore, while most text mining research concentrate on processing English documents only, mining from documents written in other languages allow access to previously unexploited information and offers a new host of opportunities. Data can be found in many different forms. Some formats are more appropriate for automatic data analysis and easier to handle than others. The usual data analysis methods assume that the data is well-defined in a number of fields with a predefined range of possible values. The question is what can be done if the data is stored purely in textual form, consisting of no records and no variables. Several document categorization techniques were developed to classify documents into pre-defined categories based on the vector-based model. The dimensions of the vector space are formed by the important words given in the documents. The documents that have already been categorized, according to the distances between

A draft version.

the vectors, are used to generate model for assigning content categories to new documents. (Mitchell, 1997) describes techniques to integrate machine learning and data mining for data analysis with varying knowledge representations and large amounts of data. (Cohen & Singer, 1996) discusses rule-based learning classifier RIPPER in the context of mail filtering. RIPPER forms sets of simple rules for data described by sets of attribute-value pairs. Each rule tests a conjunction of conditions on attribute values. Rules are returned as an ordered list, and the first successful rule provides the prediction for the class label of a new example. The system uses large batches of training data to learn the rules in a greedy fashion. The classifier must constantly be kept up-to-date and training and classification are highly intertwined since new rules are formed when a sufficient amount of data has been covered.

Another classification algorithm that provides efficient training and quick classification is naive Bayes (Hastie et al., 2001; Lewis & Ringuette, 1994; McCallum & Nigam, 1998; Mitchell, 1997). In this algorithm, adding a document to a trained model requires the recording of word occurrence statistics for that document, no rule need to be learned and no weights need to be optimized. Training consists of updating word counts and classification consists of normalized sum of counts corresponding to the words in question. Hence, training and classification are both simple and efficient and can be integrated into the learning model.

Another classification approach uses background knowledge as indices into the set of labeled training examples (Zelikovitz & Hirsh, 2001). If a piece of knowledge is close to both a training example and a test example, then the training example is considered close to the test example, even if they do not share any words. In this way, the background provides a mechanism by which the labeled examples are chosen to be used for classification of a new test example. However, these approaches neglect the explanations of why particular categories have been formed and how the different categories are related to each other. Some aspects of text mining involve natural language processing (Jackson et. al, 2002; Manning et. al, 2001) where the model of reasoning about a new text document is based on linguistic and grammatical properties of the text, as well as extracting information and knowledge from large amount of text documents.

In this paper, we focus on processing Arabic-written documents (standard and Arabian Gulf dialects) in order to classify, extract, and analyze information about fanaticism. The paper is organized as follows. Section 2 presents an experimental methodology used to collect data, it briefly discusses the origin, amount and length of raw documents. Section 3 demonstrates the way they were preprocessed and transformed into the final bag-of-words representation. Section 4 concisely summarizes the learning algorithms used to distinguish the document content. The main attention is paid to the random forest classifier, an overview of the analytical tools available when dealing with this paradigm is given. Section 4 presents and evaluates empirical results while section 5 focuses on their interpretation and future utilization. Finally, conclusions, limitations, and future work can be found in section 6.

2. The Kuwait E-mail Dataset

The dataset which was to be classified contained 300 answers to questions related to terrorist tendencies which were tagged by a domain expert. The expert assigned the responses into one of three possible classes:

- Non-Fanatic (NF) - the text does not exhibit any terrorist/fanatic tendencies.
- Code Attitude Fanaticism (CAF) - the person who wrote this text agrees to fanatic actions.
- Code Red Fanaticism (CRF) - the person who wrote this text has strong fanatic tendencies.

There were 10 questions asked of teachers and students in Kuwait, who provided their answers anonymously.

1. How do you justify the war of USA in Afghanistan?
2. Why do you think Osama Bin Laden is declaring war against USA? Would you sympathize with his ideology and actions?
3. Do you think the western world in general, and USA in particular are targeting Islam in their pursuit of fighting the war against terrorism? Explain you answer with as little words as possible.
4. Do you think the peace between Arab nations and Israel is possible? If you think not, explain why with a few words.
5. Do you justify committing suicidal actions against the adversaries in the name of holy war? Are you a supporter of such actions?
6. How do you evaluate our relations with the USA, and also the USA relations with Israel?

7. Do you see any future for having two independent countries of Israel and Palestine living in peace next to each other? If not, explain why in a few words?

8. Do you think the idea of Jihad (i.e. holy war) as a religious commandment is the main reason of violence against USA interests? Or could it also be related to political and personal ambitions of the perpetrators.

9. Would you approve any sort of violent actions against Jew or Christian civilians in the name of holy war?

10. Do you agree with the USA presence in the Gulf region? How do you see the cooperation between the Gulf countries' regimes with the USA government in fighting the war against terrorism?

The answers were short paragraphs of about 100 words for each question. The task was to classify as accurately as possible the responses and to inspect the internal structure of the dataset using mechanisms mentioned thereunder. Each single answer corresponds to a single document. The domain expert categorized the documents as follows: NF - 135 documents, CAF - 65 documents and CRF - 100 documents.

3. Bag-of-Words Representation

Raw texts represent unstructured data inappropriate for direct automatic analysis. That is why, texts have to be converted into a structured representation first. In the field of text categorization, the documents are most often represented as word-vectors usually referred to as bag-of-words. The bag-of-words representation is equivalent to an attribute-value representation as used in machine learning. Each distinct word corresponds to a feature whose value represents the number of occurrences of the given word in the document. Although this representation clearly loses information as the sequence and context of words is not considered, the words proved to work well as representation units in many tasks.

The word vector was constructed as follows. The Arabic words were translated into their English counterparts with uniform morphology first. The stopwords were removed, occurrences of all the other words were counted. Words were considered as features only if they occurred at least in 3 different documents. This process resulted in the bag consisting of 651 keywords. Most keywords appear in 4 different documents (105), on the other hand there are 3 keywords appearing in hundred and more documents (america, muslim, war). The median is 6 distinct documents per keyword. Regarding frequency of keywords, the median document is represented by 21 occurrences of keywords (the same keyword can occur more times in a single document within this statistic). Figure 1 shows (a) distribution of keywords in documents and (b) distribution of documents according to keyword occurrence.

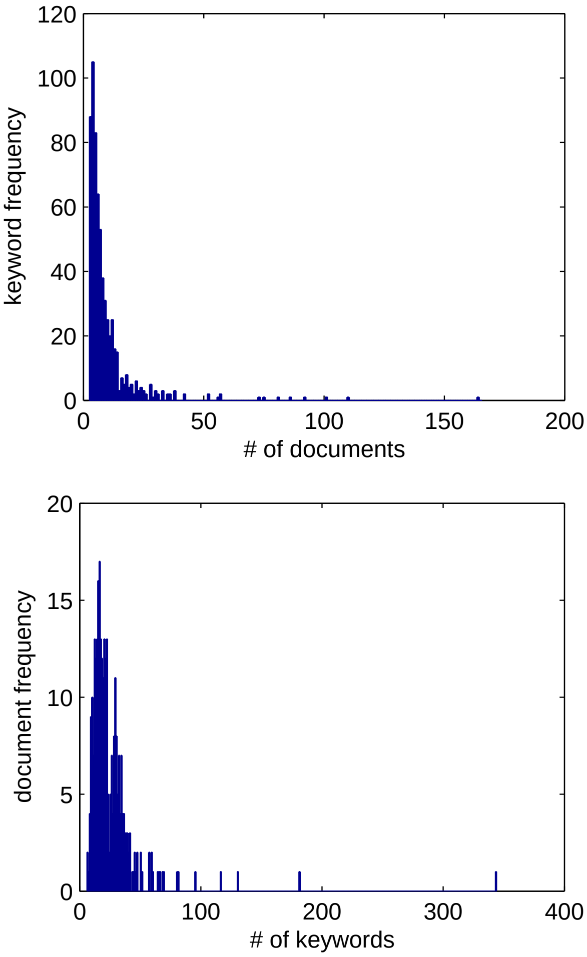

Figure 1. Relation between keywords and documents within the bag-of-words representation of the fanatic documents.

4. Learning Algorithms for Text Categorization

Text categorization dealing with the bag-of-words representation usually brings extensive and sparse data matrices. One way to avoid these high dimensional input spaces is to assume that most of the features are irrelevant and to apply feature selection prior to learning. Unfortunately, in text categorization there are often only very few completely irrelevant features (see e.g., the experiment in (Joachims, 1998)). Consequently, a good classifier should combine many relevant features (learn a dense concept) since aggressive feature selection may result in a loss of information.

In this text we apply several well-known learning algorithms for text categorization. However, the main emphasis is paid to application of a random forest classifier. A random forest is a tree ensemble created in a way combining well-known idea of bagging with random feature selection. Random forests proved to be an effective tool in classification and prediction. They also provide introspective measures such as variable importance, proximity (clustering), out-of-bag Bayes error estimates, which can be used to discover internal structure of a dataset.

4.1. TFIDF Classifier

The first algorithm which can easily be applied for text categorization is TFIDF classifier. The classifier is based on the relevance feedback algorithm for information retrieval. Learning is achieved by combining the term frequency inverse document frequency (TFIDF) vectors (Salton & Buckley, 1997) into a class prototype vector. A new document is classified into the class which maximizes the cosine distance between the vector of document to be classified and the class prototype vector. We also use the probabilistic counterpart of the algorithm denoted as PrTFIDF. This modification was introduced in (Joachims, 1997). In this paper, we use our own implementations of the above mentioned algorithms written in Python.

4.2. Multinomial Naive Bayes Classifier

Naive Bayes text classifiers have been widely used because of their simplicity. Among the various versions of classifiers, multinomial naive Bayes (mNB) text classifier (McCallum & Nigam, 1998) is mostly used. This type of model, which is also known in statistical language modelling for speech recognition as "unigram language model", usually performs well even for larger vocabulary sizes. A document is understood as an ordered sequence of word events drawn from the vocabulary. The probability of each word event in a document is independent of the word's context and position in the document (naive Bayes assumption). Then, each document is drawn from multinomial distribution of words with as many independent trials as the length of the document. The WEKA implementation of this classifier was applied (weka.classifiers.bayes.NaiveBayesMultinomial, see (Witten & Frank, 2000) for details).

4.3. Support Vector Machines

Support Vector Machines (SVMs), introduced by Vapnik (1995), are based on statistical methods minimising the risk of error and offering solutions to optimise generalisation performance. The idea is to find a hypothesis for which we can guarantee the lowest true error. The true error of the hypothesis is the probability that it will make an error on an unseen and randomly selected test example. An upper bound can be used to connect the true error of the hypothesis with its error on the training set and the complexity of the hypothesis space (measured by VC-Dimension). SVMs find the hypothesis which (approximately) minimizes this bound on the true error by effectively and efficiently controlling the VC-Dimension of the hypothesis space. SVMs are capable of overcoming the problems associated with high dimensional spaces (e.g. overfitting). The WEKA implementation of this classifier was applied (weka.classifiers.functions.SMO).

4.4. Random Forest Classifier

Random forests (RF) are a relatively new promising classification method proposed by Leo Breiman (2001). The random forest method combines individual decision trees into large ensembles. Classification is quite straightforward: each tree in the forest casts a vote for a particular class and the most popular class is selected as the output of the RF. Each tree has the same weight in voting (fair voting scheme). Random forests can deal with high dimensions and distinguish feature relevance which makes them a suitable tool for text categorization. In classification accuracy, random forests are competitive with today's best classifiers such as the support vector machines. Random forests however provide some interesting insights which will be discussed in the following paragraphs. The fact that an RF classifier may be readily decomposed into its individual components has many practical uses:

- Variable importance - the contribution of each variable to the overall classification performance is computed.
- Proximity computation - RFs generate a similarity measure of the individual examples in the dataset.
- Out-of-Bag (OOB) error - a by-product of the training process, the OOB error provides an unbiased estimate of the generalization error.
- Outlier detection - may be computed from the proximity data.

In this paper we deal with the randomForest library of R package. In the following paragraphs, the focus will be on the use of Random Forests in supervised learning applications. The proximity measure, variable importance and the OOB error will be further explained.

4.4.1. Training a Random Forest

Random forests are trained by inducing a predetermined number of decision trees. The trees are induced individually and independently which allows for simple paralellization if necessary. Each tree is trained according to the following algorithm:

1. The training set (of cardinality N) is resampled with replacement to generate a new training set with N examples (boosting).

2. A decision tree is grown to the maximum extent possible. At each split, instead of computing the criterion for each possible split, pre-select F random variables and select the best split using the criterion. No pruning should be performed either before or after the training.

The training process has basically two parameters pertinent to RF: a size of the Random Forest (number of trees) and a number of variables to test at each split (F). (Breiman, 2001) recommends to set the value of F to $\log_2 M$, where M is the number of variables of the dataset. There is no upper bound on the number of trees induced, however there is a minimum number of trees that should constitute the RF, so that its statistical properties hold. Other parameters, such as the split criterion, split type for nominal variables, etc. pertain to the method of constructing individual decision trees and is not considered as a parameter to the RF method itself. The RF method exhibits good properties with respect to both of the variables noted above. RFs do not overfit even with a large number of trees, i.e., generally the performance of the classifier improves (albeit at slower rates with larger ensembles) as the number of trees is increased. The process is quite insensitive to the value of F on a large range, thus further stabilizing the performance. This is further clarified in the paragraph on RF theory. Some datasets are badly balanced (some classes are much more populated than others) and this has an adverse effect on the classifier accuracy in identifying those classes. RFs provide a method to balance the dataset by assigning weights to each example. The weights affect the way the dataset is resampled for each tree. The examples with higher weight have more chance of being selected multiple times, thus in effect behaving as if there were multiply present.

4.4.2. Out-of-Bag error

The training process has a useful by-product: the out-of-bag error. On average about 1/3 of the examples of the original dataset are not used in the construction of any particular decision tree. These examples are considered OOB for the particular decision tree. It is then possible to use these unseen examples to validate the training of the induced tree. (Breiman, 2001) states that the OOB error has been empirically shown to produce an unbiased estimate of the generalization error. The OOB may be computed during the training process with minimal additional cost, which provides the experimenters with an estimate of the performance immediately after completion of the training process.

4.4.3. RF Theory

RF theory examines two important quantities related to the internal workings of the RFs: strength and correlation. The performance of the RF improves with strength of individual trees but is decreased with their mutual correlation. The problem is that they are locked together in one degree of freedom of the training process (the parameter F - random variables considered at each split). The strength of a classifier is a measure of its performance or quality, the higher the strength, the better the expected performance (generalization error criterion). Intuitively, it may be stated that the strength of the decision tree increases as the parameter F increases (there is a better chance of finding the global minimum of the criterion function if there are more tries). Increasing F also increases the correlation between the trees. The correlation is a measure of similarity between the trees in the ensemble. A higher correlation has the effect of increasing the upper bound on the generalization error. Intuitively, as the parameter F increases, the trees are more likely to take similar decisions concerning splitting. As noted above, these two quantities are balanced for a wide range of the parameter F. For formal treatment of the two quantities, see (Breiman, 2001).

4.4.4. Proximity Measure

A trained RF may be used to compute a similarity measure relating the examples to each other. This similarity measure is in the form of a matrix of proximities. This matrix may be transformed into a distance matrix and further processed for example by multidimensional scaling (MDS) to create images of the structure of the dataset. In our problem, this method has been used to discover an intrinsic sub-class structure in the dataset. The following algorithm serves to compute the proximities in a straightforward manner (assuming a trained RF is already available):

1. Run all examples down a tree and remember at which leaf node the example terminated. If any two examples terminate at the same leaf node, increase the proximity between them (i.e. in a proximity matrix, increase prox(i, j) and prox(j, i) if the examples were the j-th and i-th).
2. Repeat the above procedure for all trees.

After all of the trees and examples have been processed, the matrix contains elements which measure the closeness of all pairs of examples with respect to the trained trees.

4.4.5. Variable Importance

One disadvantage of classification methods that are based on ensemble of trees is the loss of their immediate interpretability. However, the random forests can interpret their outcome in terms of variable importance. The variable importance is a measure of the relevance of a variable to the overall classification performance of the random forest classifier. RFs are particularly suited to measure the variable importance, the methodology is as follows:

1. Classify a dataset and determine the classification error (OOB).
2. Noise up or permute (recommended) the values of the i-th variable and repeat the classification again. The difference of the classification errors is the variable importance.
3. Repeat the previous step for all the variables in the dataset.

It is recommended to permute the values rather than introduce noise as the noise is not likely to preserve the distribution of the particular variable thus giving biased results. The classification error can also be replaced by a so-called margin, which is more sensitive to changes caused by permuting of the given variable. The margin is defined as the proportion of votes for the correct class minus the maximum proportion of votes for the other classes.

5. Experiments and Results

Table 1. Fanatic text classification - accuracy of the learning paradigms.

Algorithm	Accuracy	Better wrt
TFIDF	59.0	×
PrTFIDF	62.7	×
mNB	62.0	×
SVM	62.0	×
RF	69.0	√ all

In the first instance this section provides empirical tests on performance of the learning algorithms mentioned in the previous section. The goal is to distinguish the degree of fanaticism (NF, CAF, CRF) within the set of 300 documents annotated by a human expert. The documents are represented in the bag-of-words format as described in Section 3. The problem is understood as a classification task, the aim is to maximize the classification accuracy, i.e., to match human annotations as frequently as possible. In order to estimate generalization error, leave-one-out cross-validation (LOOCV) is used. The uniform testing procedure was followed for all the learning paradigms, in case of parametric methods (SVM, RF) the default parameter setting was used. Performance of the classifiers is mutually compared by the McNemar's test which is referred to as the only test with acceptable type I hypothesis testing error (Dietterich, 1998) (p value was set to 0.05). The results are presented in Table 1.

The classification accuracies suggest that the degree of document fanaticism cannot be reliably categorized by any of the tested algorithms. There is a twofold reason for this. First, the documents we deal with are relatively short. They can often be too brief to express the potential fanaticism clearly and even human annotations can be considered as fuzzy recommendations rather than definite statements. Second, the selected bag-of-words representation can miss "subtle" semantic distinctions that finally express and indicate content fanaticism. For example, negation (or the words 'no', 'not', etc.) is not included in the extracted features. Therefore if the word *peace* were used in a negative sense (i.e. 'There cannot be peace until ...') there would be no difference in the feature vector.

On the other hand, all of the classifiers can be considered as informed as they significantly outperform the trivial classifier assigning the majority class to all of the documents (the trivial classifier assigns all the documents the non-fanatic class and reaches 45% accuracy). What is more, the classifiers mostly tend to misclassify the halfway code-attitude fanaticism class. When used e.g., to alarm for potential code-red fanaticism documents their precision and recall are considerably higher than the above-reported accuracy - RF classifier shows about 82% recall and 72% precision. There is only 15% error when considering NF-CRF misclassifications. In other words, there are only 36 documents (out of 235 documents belonging to NF and CRF classes) falling outright into contrary class and there is only one more document falling into CAF class.

When mutually comparing the learning algorithms, the results seem to be similar except for the random forest classifier. When applying the McNemar's test with p=0.05, the null hypothesis that the RF classifier shows the same accuracy as a competitive classifier can always be rejected in favor of the alternate hypothesis that the RF classifier is better. As usually, there is a performance-complexity trade off as the RF ensemble classifier is the most memory and computationally demanding. Nevertheless, it represents a learning paradigm that provides both good performance and insight into its decision making process.

6. Insight into RF classifier

Besides pure classification, the task of automatic categorization of fanatic texts can also be viewed as a descriptive task. We can ask questions such as:

- Are there any words whose mere occurrence within a document signifies its fanaticism (to a human expert at least)?
- Is there any internal document structure? Can we further portion the existing classes into more compact document clusters which would be easier to describe?
- Or generally, can the current classifiers help when developing a prospective better (more informed) representation of the domain?

We picked the most accurate classifier (RF) supposing that it best models the domain and further analyzed it. Random forest is not a simple model on any account - our model consists of 1000 trees, one of 25 randomly selected variables can be selected at each split. But, the model does not have to be simple to provide insight and interpretability, e.g., reliable information about the relation between predictor and response variables. Section 4.4 gives theoretical ways of getting information from forests. This section applies these ways in the domain of fanatic documents.

6.1. Keyword Importance

Relevance of keywords to the overall classification performance of the random forest classifier can be studied. The ordering shall not serve for the feature selection purposes primarily as the weakly relevant features can still improve automatic decisions. However, they can help to characterize the individual classes as well as to develop a more profound representation (linguistic phrases, non-consecutive phrases, explanation patterns).

Within the fanatic domain we have used two different criteria: the mean decrease in accuracy over all classes and the mean decrease in node impurities from splitting on the keyword. The second criteria was measured by Gini index - for details on Gini index see (Breiman et al., 1984). Let us remind that both the criteria confront the values reached with the original and permuted keyword frequencies - the higher the decrease the higher the importance.

The results are summarized in Figure 2, the most important keywords appear at the top of both lists. At first sight it is clear that the keyword frequency makes the necessary condition for its importance. The most important keywords must appear in a large number of documents - the top ten keywords appear in 20 documents at least (while median for all the keywords makes 6 documents). Nevertheless, the keyword ordering does not follow the frequency ordering exactly and therefore it brings additional information.

The class-specific measures computed again as mean decrease in accuracy can also be evaluated. For example, the CRF class is best discriminated by keywords: *terrorise*, *land* and *fight*. Interestingly, occurrence of *terrorise* signifies a non-CRF document as opposed to the other two keywords mentioned-above.

RF learners represent a stochastic learning technique. That is why it is also vital that the keyword importance values are stable in magnitude as well as in the order. They do not show a high variance with changing random seed nor critical forest parameters.

6.2. Proximity

Interesting results were obtained computing the proximity matrix. Figure 3 confirms the difficult separability of the individual classes discussed in Section 5. The classes tend to overlap widely, which affects mainly the halfway class documents located just in between the clusters of the remaining classes. However, this allocation corresponds well to the reality - code-attitude documents are truly expected to bridge the gap between non-fanatic and code-red documents.

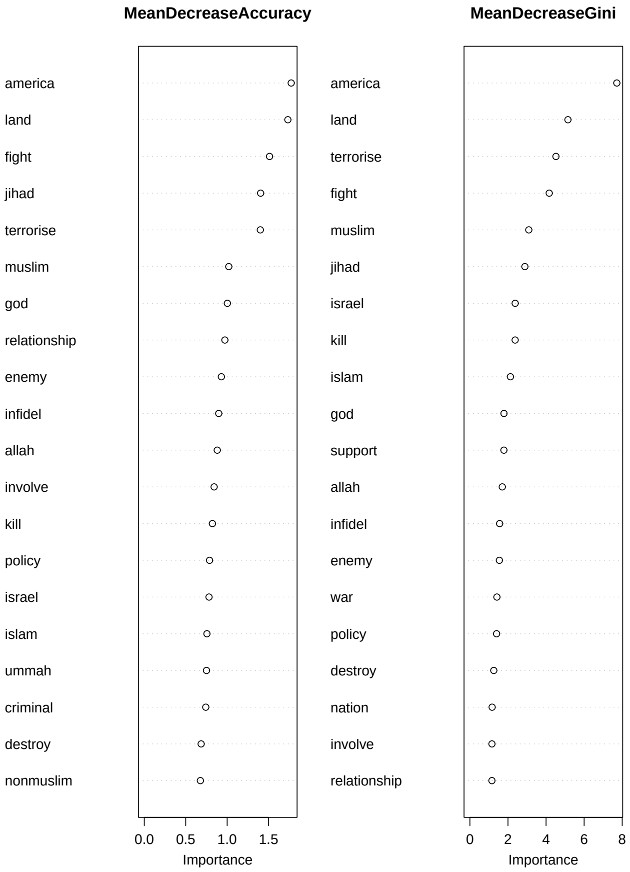

Figure 2. Keywords sorted according to their significance with respect to classification accuracy and Gini entropy criterion.

Another interesting point is the structure of document clusters. Disregarding the human annotation, there appear to be two clusters clearly visible in the image. The distinction is most apparent for the non-fanatic class (light gray triangles), there are clearly two separate clusters - the upper-right and the bottom-center one. This means that the trained RF identified a certain internal difference, which can be best observable within the non-fanatic class. As the only information contained in the selected representation is a word frequency in each answer, thus it is reasonable to suspect that the two subgroups differ largely in the distribution of certain words.

Let us define two clusters consisting of NF documents entirely as seen in Figure 3. The issue of terrorism seems to make the difference between the clusters. It is compelling that the word *terrorise* appears 96 times within the first cluster documents while it occurs in no document of the second cluster. The first cluster is more likely to contain words such as: *act, attack, crime, ideological, islam, jihad, laden, life, movement, muslim, nonmuslim, religion, violate* or *terrorise* already mentioned earlier. The second cluster tends to comprise words such as: *east, economy, force, future, iraq, middle, regime, sadam, unite* and *weapon.* The first cluster apparently contains the documents discussing issues of religion and terrorism, while the second one is more focused at secular issues which can be typically represented by the regime of Saddam Husajn.

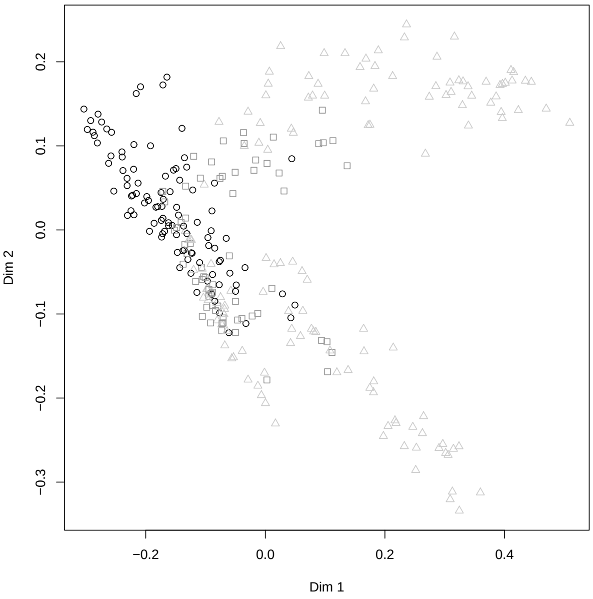

Figure 3. Proximity image with 2 underlying dimensions, NF documents are in triangles, CAF in squares and CRF in circles. Fanaticism is also distinguished in shades of gray - the darker the more fanatic.

The results confirm the truism that it is much easier to distinguish the topic of the document - the clusters are well pronounced - than the writer's attitude - the classes tend to overlap. Nevertheless, the classifier often gives an informed guideline what is the actual degree of fanaticism within the document.

7. Conclusion

This paper discusses the task of automatic categorization of fanatic texts. The texts are supposed to be split into three distinct categories (non-fanatic, code-attitude and code-red) in accordance with formerly known human expert annotations. In order to solve the task, the bag-of-words representation was combined with various attribute-valued classifiers. The experiments revealed that the most reliable categorization can be reached with the random forest classifier. Moreover, the RF classifier can also provide an insight into its decision making process. Although the degree of fanaticism cannot be consistently distinguished, the classifier is potentially valuable to alarm for code-red fanaticism documents (82% recall and 72% precision).

The presented procedure represents the most straightforward way to solve the task and there is still a large room for improvements and future work. First, it is advisable to operate with more representative and extensive set of documents. The current set of documents is a result of a rigid questionnaire applied to a diverse but still limited group of people. There is almost an unlimited potential of documents available via Internet, which can be an excellent source for diverse documents approaching the future classifier to its prospective area of application. Second, the documents should be represented in a more sophisticated way than as yet. Section 6 may give an initial clue in development and application of more semantically and problem-oriented representation (phrases, ontology, explanation patterns etc.).

Acknowledgments

References

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees.* Monterey, CA: Wadsworth and Brooks.

Cohen, W. W., & Singer, Y. (1996). Context-sensitive learning methods for text categorization. *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval* (pp. 307–315). Zürich, CH: ACM Press, New York, US.

Dietterich, T. G. (1998). Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, *10*, 1895–1923.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2001). *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations.* New York: Springer-Verlag.

Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. *Proceedings of ICML-97, 14th International Conference on Machine Learning* (pp. 143–151). Nashville, US: Morgan Kaufmann Publishers, San Francisco, US.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. *Proceedings of ECML-98, 10th European Conference on Machine Learning* (pp. 137–142). Chemnitz, DE: Springer Verlag, Heidelberg, DE.

Lewis, D. D., & Ringuette, M. (1994). A comparison of two learning algorithms for text categorization. *Proc. Symposium on Document Analysis and Information Retrieval SDAIR-94* (pp. 81–93).

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification.

Mitchell, T. M. (1997). *Machine learning.* New York: McGraw-Hill.

Salton, G., & Buckley, C. (1997). Term-weighting approaches in automatic text retrieval. 323–328.

Vapnik, V. (1995). *The nature of statistical learning theory.* New York: Springer.

Witten, I. H., & Frank, E. (2000). *Data mining: practical machine learning tools and techniques with java implementations.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Zelikovitz, S., & Hirsh, H. (2001). Using LSI for text classification in the presence of background text. *Proceedings of CIKM-01, 10th ACM International Conference on Information and Knowledge Management* (pp. 113–118). Atlanta, US: ACM Press, New York, US.