Automated News Categorization using Machine Learning methods

U Suleymanov[1] and S Rustamov[2,3]

[1] State Agency for Public Service and Social Innovations under the President of the Republic of Azerbaijan, Baku, Azerbaijan
[2] School of Information Technologies and Engineering, ADA University, Baku, Azerbaijan
[3] Institute of Control Systems of ANAS, Baku, Azerbaijan

Email: srustamov@ada.edu.az, ADA University

Abstract. Being one of the most linguistically rich languages, Azerbaijani has been researched less in the context of natural language processing area. The text corpus created from Azerbaijani news articles is designed to apply supervised machine learning approaches for the case of automatic news labeling. Chi-squared test and LASSO methods have been implemented for feature selection and pre-processing. The application of supervised machine learning approaches to the text corpus allowed us to compare the performance results of well-established supervised machine learning approaches in the domain of Azerbaijani language.

1. Introduction

In text classification, different statistical and machine learning methods are applied in order to automatically assign one of the predefined labels to a given element of the unlabeled document space. More formally, if some d is a document within the entire set of documents D and C = {c1, c2, c3, …, cn} is the set of all the categories, the text classification assigns one category ci to the given document d. Classifier γ trained on a training set D of labeled documents, determines the label of the previously unseen document according to the learned parameters. The name of supervised learning is obvious from the above definition, namely, machine learning models learn best parameters for predication under the supervision of training data set [1].

Text classification is not a linear process rather it requires iterative approach. Namely, for the predictive model to be able to learn optimal parameters and to predict accurate results, the statistical trends and patterns observed during predictive model training phases should be applied to data preprocessing and vice versa. Iterative approach contributed a lot to the development of news corpora and to the determination of optimal categories under which news articles have been grouped.

Feature extraction is the first and one of the integral parts of the text classification. Traditional machine learning models cannot work with textual data, and in order for them to be able to learn parameters and make predictions, numerical features of the document should be extracted. Bag of words is a general approach taken for feature extraction [1]. The feature space generated by bag of words approach, especially for bigrams and n-grams are very huge and sparse. Therefore, for enabling machine learning models to work with this type of high dimensional features and for optimization of model performance, the selection of essential features becomes priority [2].

Text classification problem can be solved by a number of machine learning approaches such as support vector machines, artificial neural networks, decision tress and etc. Text classification problems are distinguished by their high dimensional feature space from other machine learning problems. Therefore, algorithms which can work with thousands of features efficiently performs best in the text classification.

2. Related work

Text classification begins with feature extraction. Application of different future extraction methods yields different performance results even if we apply the same classification algorithm in the next step. Traditional approach to feature extraction is bag of words where each word is represented as a feature [1]. Bag of words approach does not take the order of words into account and represents the count of each word in the document as feature [1]. Term-frequency inverse-document-frequency approach of bag of words weights documents not only by their counts but also by their frequency in the whole training dataset. This allows to weight some words such as ‘and’ appropriately as these types of words are common in whole dataset but carries little semantic information about the actual category of the document.

Feature selection is next step in text classification which helps to distinguish and extract relevant features from the huge feature space. Researchers have developed several feature extraction methods in order to feed classifiers with necessary feature set and to decrease error rate. In filter based feature selection schemas features are evaluated and assigned weights which at the last step are taken as selection criteria [3]. Least absolute shrinkage and selection operator (LASSO) is another approach implemented widely for feature selection as it penalizes the coefficients of features in the cost function [4]. This penalization makes coefficients of features approach zero and for features which are weakly related to document category the coefficients become zero if penalization parameter is chosen adequately [4].

Text classification has been approached by a number of classification methods by researchers. Naive Bayes classifier is based on Bayes theorem and tries to solve the classification problem by probabilistic approach. [5]. Naive Bayes approach assumes that there is no correlation between features in other words they are independent. In spite of the fact that words in the given document are semantically connected, Naïve Bayes approach still performs reasonably well for text classification [5].

Another approach generally applied to text classification is Support Vector Machines [6]. Joseph (2015) analyzes the application of Support Vector Machines with word2vec and tf-idf approaches.

Artificial neural networks are researcher’s choice of classifiers in recent years. Lai (2015) presents the application of recurrent convolutional neural networks and its comparison with other well established text classification methods. Recurrent convolutional neural networks can learn contextual information from text and outperforms other classifiers in several datasets [1]. Several novel approaches have been implemented for artificial neural networks as well. As seen in [7], [8] and [9] neural fuzzy systems and their hybrids have been implemented successfully for sentiment analysis.

3. Data preparation and feature extraction

As classification algorithms work with numerical data, it is necessary to handle textual data properly and extract relevant features from it. Moreover, having deep understanding of dataset enables applying more suitable feature extraction and selection techniques to classification problem. We will discuss different feature extraction and selection methods. In the classifiers section we will provide the analysis of applying different feature extraction methods for naive Bayes, support vector machines (SVM) and artificial neural networks classifiers.

3.1. Dataset

In order to be able to apply feature extraction as well as machine learning approaches 130000 news articles have been gathered along with their assigned categories. The documents are grouped under 8 mutually exclusive categories. This allows us to train and test different text classifiers and build a predictive model. Moreover, the dataset has been splitted randomly into train and test sets so that the accuracy of classifiers can be determined.

3.2. Count vectorization

For the machine learning methods to be able to learn, the features have to be extracted from documents. Bag of words approach considers each unique word as feature and expresses each document as the set of features. By this way feature space has as many dimensions as the number of unique words in the whole dataset. Count vectorization method assigns the count of the word in the document to its corresponding feature. Therefore, count vectorization is not able to represent word orderings or specific word combinations. As most of the words in the whole dataset dictionary is not present in the specific document, most of the feature values are zero for given document.

3.3. TF-IDF vectorization

Term frequency inverse document frequency approach not only considers the frequency of the given word but also weights it inversely by the word's frequency in the whole dataset. This allows to reduce the weights of the common words which is not influential for the category of the document [10]. Term frequency inverse document frequency method is able to weight common words by its IDF part:

$$idf(t) = \lg \frac{1+n_d}{1+df(d,t)} + 1 \tag{1}$$

In equation (1), t is the term in the given document d, while n_d and $df(d, t)$ expresses the whole document count and the number of documents containing t respectively. Thus, if a word is frequent in most of the documents, the denominator and numerator gets close to each other and IDF score approaches zero. Thus words which is not discriminative enough get close to zero weights [10]. We have applied both types of feature extraction methods and trained two classifiers naive Bayes and SVM. Comparative analysis of these methods for the SVM classifier shows that SVM classifier has 2.5 % more accuracy while working with features generated by tf-idf vectorization method rather that count vectorization.

3.4. Stop word removal

Classifier assigns different weights to each word while defining the category of the document. For instance, the weight assigned to the word "aktyor" (actor) for economics category should be small, while its weight for art category should be high. However, some words in the document are so common that they do not play any role in determining document category. The words "üçün" (for) and "hər" (each) are examples of such words. These types of words are commonly referred as stop words and removing them generally benefits the classification accuracy as this reduces the feature space. We have applied the stop word removal technique and get 0.4% improvement in accuracy for support vector machines classifier. The same technique for naïve Bayes classifier gave 0.2% improvement.

4. Feature selection

4.1. Chi-squared

The feature selection techniques reduce the burden of the classifiers in terms of choosing relevant features for classification. Namely, classifiers have been shown to perform better when they are provided with less and more important features. Chi-squared is a well-known statistical approach to measure the relation between each feature and target class. By taking the chi-squared test values of each feature and ordering them by descending order, we can select n features with highest values. Extracting relevant features, we got 88.6% accuracy with SVM which is 0.7% more than the accuracy result of SVM with tf-idf vectorization.

4.2. LASSO

Least absolute shrinkage and selection operator is a well-known approach to feature selection. It applies the following l1 penalty to ordinary minimized sum of squares method [11]:

$$\lambda \sum_{i=1}^{N} |\beta_i| \tag{2}$$

N denotes the number of features and λ denotes the regularization parameter. Choosing high values of λ makes the weights assigned to features go to zero. If we choose the parameter appropriately, only the features relevant for the classification have coefficients different from zero. In our case, extracting features with LASSO increased the accuracy of naïve Bayes classifier from 71.5% to 76.9% while it increased the SVM classifiers accuracy from 87.9% to 88.5%.

5. Classifiers

5.1. Naïve Bayes

Firstly, we trained the naive Bayes classifier with count vectorization method. Bayes rule is at the core of naïve Bayes Classifier. In order this rule to work, it is assumed that features are independent of each other. As words are connected semantically in news articles, this assumption does not hold true. Regardless of that, for text classification problems naïve Bayes classifiers gives reasonable accuracy results. For instance, on famous 20 Newsgroups and WebKB datasets, it shows 85% and 86% prediction accuracy respectively [12].

By applying naive Bayes, the highest accuracy we got was 80.4%. This accuracy is achieved by applying count vectorization as feature extraction. The least accuracy observed was by using tf-idf approach with Naïve Bayes as seen in Fig.1. We can also observe from figure 1 that the most successful feature selection approach was chi-squared test with accuracy result of 79.6%.

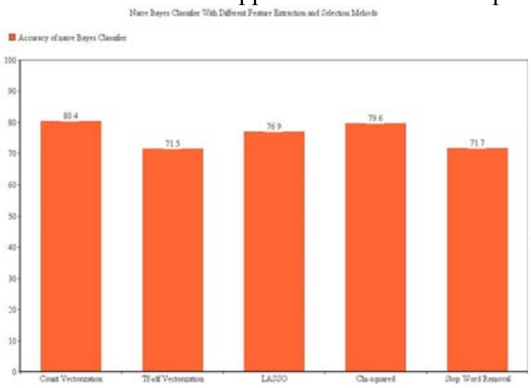

Figure 1. Accuracy of naive Bayes classifier with different feature extraction and selection methods.

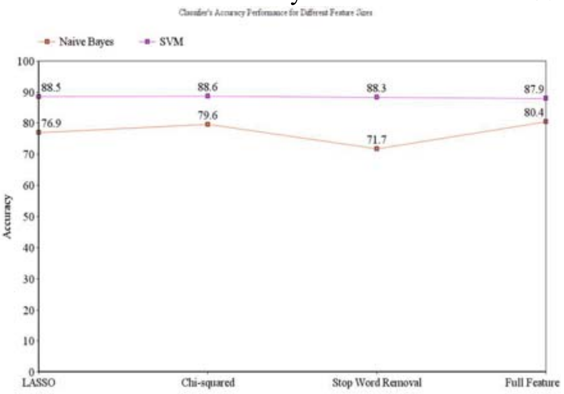

Figure 2. Accuracy of naive Bayes and SVM classifier with different feature sizes.

Applying different feature selection methods produce different feature set as well as feature sets with different sizes. In our case minimum feature space size is generated by LASSO while maximum was generated by applying just feature extraction without using feature selection. We decided to measure the effect of feature space size on classification accuracy. Figure 2 demonstrates naïve Bayes and SVM classifiers' accuracy measures with different feature space sizes.

It should be noted that in case of applying no feature selection, we took the maximum accuracy results of count and tf-idf vectorizer as they have the same feature space size. Upon analysing figure 2 we can come to the conclusion that for our case chi-squared performed best amongst feature selection methods.

5.2. Support vector machines

Applying support vector machines (SVM) was the next goal for us. The prediction accuracy of SVM for other datasets is shown to be very high as it can handle classification problems with high feature space sizes. For our case also, SVM showed high performance results. Figure 3 describes SVM classifier's accuracy in different data pre-processing settings.

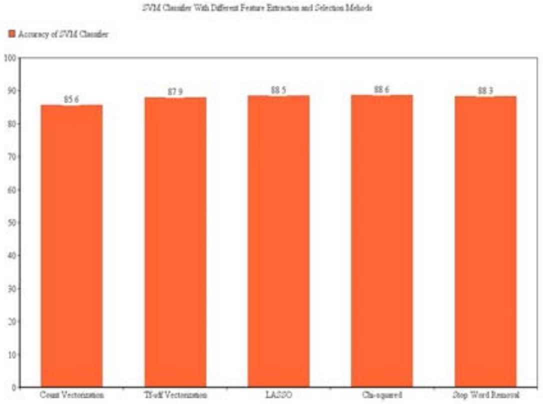

Figure 3. Accuracy of SVM classifier with different feature extraction and selection methods.

As figure 3 suggests SVM classifier performs reasonably well in all feature extraction and selection settings. However, we saw the greatest increase in accuracy when we changed the feature extraction approach from count to tf-idf.

5.3. Artificial neural network

Artificial Neural networks have been implemented in various settings and for solving a variety of problems. Text classification is amongst them and artificial neural networks is proven to perform well in this domain [13]. Artificial neural networks can be arbitrarily complex in terms of architecture. Therefore, getting highly accurate results can be computationally expensive. In order to make the calculations less expensive and to make the architecture more compact, the feature selection is a reasonable approach in text classification as it enables to deploy less and more influential features to the artificial neural network. Artificial neural network model gave 86.3% accuracy result on Azerbaijani news article dataset. Application of feature selection namely, Chi squared test increased the accuracy of artificial neural networks by 2.8%.

6. Conclusion

Because of the dataset being new, comparing the results obtained by implementing various pre-processing methods with other benchmarks is difficult for us. Therefore, we tried to compare the accuracy results of classifiers with the accuracy results obtained on famous datasets. Moreover, the dataset can be made available to other researchers if they request it by contacting us via email.

As shown in the classification accuracy of naïve Bayes and SVM, feature extraction and selection is an important step for text classification. Moreover, it is impossible to say which pre-processing approach is best, because of two factors. Firstly, choosing pre-processing approach depends heavily on dataset properties such as size, the length distribution of documents within dataset and etc. Secondly, even if we keep dataset constant, the different classifiers performs differently with the features provided by various pre-processing approaches. For example, if we compare text vectorization approaches for naïve Bayes and SVM, it becomes apparent that using count vectorization yields best accuracy result for naive Bayes, while for SVM count vectorization yields lowest accuracy.

Acknowledgement
This work has been carried out in Center for Data Analytics Research at ADA University.

References

[1] Siwei L, Liheng X, Kang L and Jun Z 2015 Recurrent Convolutional Neural Networks for Text Classification *AAAI'15 Proc. of the Twenty-Ninth AAAI Conf. on Artificial Intelligence* (Austin: AAAI Press) pp 2267–73

[2] Girish C and Ferat S 2014 A survey on feature selection methods *Computers & Electrical Engineering* **40** 16–28

[3] Alper K U 2016 An improved global feature selection scheme for text classification *Expert Systems with Applications* **43** 82–92

[4] Fonti V and Belitser E 2017 Paper in Business Analytics Feature Selection using LASSO

[5] Jain A and Mandowara J 2016 Text Classification by Combining Text Classifiers to Improve the Efficiency of Classification *International Journal of Computer Application* **6** 1797–2250

[6] Lilleberg J, Zhu Y and Zhang Y 2015 Support vector machines and word2vec for text classification with semantic features *IEEE 14th Int. Conf. on Cognitive Informatics & Cognitive Computing (Beijing)* pp 136–140.

[7] Aida-zade K, Rustamov S, Clements M and Mustafayev E 2018 *Recent Developments and the New Direction in Soft-Computing Foundations and Applications* vol **361** pp 63–70

[8] Rustamov S 2018 A Hybrid System for Subjectivity Analysis *Advances in Fuzzy Systems* vol **2018**

[9] Rustamov S, Mustafayev E and Clements M A 2018 Context Analysis of Customer Requests using a Hybrid Adaptive Neuro Fuzzy Inference System and Hidden Markov Models in the Natural Language Call Routing Problem *Open Engineering* vol **8** pp 61–68

[10] Neto J L, Santos A D, Kaestner C A A, Alexandre N, Santos D, Celso A A, Kaestner A , Freitas A A and Parana C 2000 Document Clustering and Text Summarization

[11] Urda D, Franco L and Jerez J M 2017 Classification of high dimensional data using LASSO ensembles *IEEE Symposium Series on Computational Intelligence (Honolulu)* pp 1–7

[12] Aida-zade K and Rustamov S Learning 2016 User Intentions in Natural Language Call Routing Systems *Recent Developments and New Direction in Soft-Computing Foundations and Applications* pp 37–46

[13] Rustamov S, Mustafayev E and Clements M A 2013 Sentiment Analysis using Neuro-Fuzzy and Hidden Markov Models of Text 2013 *Proc. of IEEE Southeastcon(Florida)* pp 1–6