Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 8, No. 1, March 2020, pp. 178~188
ISSN: 2089-3272, DOI: 10.11591/ijeei.v8i1.1696

Persian Text Classification using naive Bayes algorithms and Support Vector Machine algorithm

Naeim Rezaeian[1], Galina Novikova[2]
[1,2] Information Technologies Department, Peoples' Friendship University of Russia (RUDN University), Moscow, Russian Federation

Article Info

Article history:

Received Oct 17, 2019
Revised Feb 3, 2020
Accepted Mar 5, 2020

Keyword:

Text classifications
Gaussian Naive Bayes
Multinomial Naive Bayes
Bernoulli Naive Bayes
TF-IDF
SVM

ABSTRACT

One of the several benefits of text classification is to automatically assign document in predefined category is one of the primary steps toward knowledge extraction from the raw textual data. In such tasks, words are dealt with as a set of features. Due to high dimensionality and sparseness of feature vector results from traditional feature selection methods, most of the proposed text classification methods for this purpose lack performance and accuracy. Many algorithms have been implemented to the problem of Automatic Text Categorization that's why, we tried to use new methods like Information Extraction, Natural Language Processing, and Machine Learning. This paper proposes an innovative approach to improve the classification performance of the Persian text. Naive Bayes classifiers which are widely used for text classification in machine learning are based on the conditional probability. we have compared the Gaussian, Multinomial and Bernoulli methods of naive Bayes algorithms with SVM algorithm. for statistical text representation, TF and TF-IDF and character-level 3 (3-Gram) [1,2] were used. Finally, experimental results on 10 newsgroups.

Corresponding Author:

Naeim Rezaeian,
Information Technologies Department,
Peoples' Friendship University of Russia (RUDN University),
Moscow, Russian Federation
Email: Naeim.rezaeian@hotmail.com

1. INTRODUCTION

With the advent of information technology, organizations and companies are increasingly turned to the Internet to transfer their information. Given that about 80% of the information is in the form of text, companies need data retrieving and mining tools to keep up with their rivals and compete through their achieved information at the right time and low cost. Data mining deals with data analysis to discover hidden and useful information about the data in the database. Text mining is an important part of data mining that organizes a set of a large text documents to capture their hidden knowledge. This science includes the classification of texts, extraction of relationships, entities, and events that are widely used in data retrieval to organize documents. Automatic document classification means assigning textual documents to predefined categories. Although text classification is studied since 1960, significant progress has been made in this regard since the early 1990s thanks to software and hardware improvements [3]. Text classification is performed in a wide range of fields including electronic mail filtering, document classification, word disambiguation, etc. There are mainly two classification approaches to enhance the organizational task of digital documents. First is the supervised approach, which is commonly used where a pre-defined category is labelled and assigned to a document based on its contents. Text categorization systems classify new documents into one label that is determined by predefined categories. Second is the unsupervised approach, which is also applied where there is no need for human intervention or labelled documents at any point in the whole process [4]. Many supervised

learning algorithms have been applied to the area of text classification, using pre-classified training document sets. Those algorithms, that used classification, include K-Nearest Neighbor (K-NN) [5,6] classifier, Naïve Bayes (NB), decision trees, Support Vector Machines (SVM) [7,8] and Neural Networks. A Bayesian method is one of the most used methods in classifying texts. In this method, the text is presented assets of independent words from each other and the location of the text. The probability function of each text is derived by the product of the probability of its words and the probability of a text occurrence with that length. System learning is done by estimating the parameters for generating a model that only uses the tagged texts. The algorithm uses the estimated parameters for new text classification by calculating which category is mostly similar to the given text. This algorithm will be discussed in detail in the methodises section. In this research, three methods from the Bayesian classifiers family, Gaussian Naive Bayes, Multinomial Naive Bayes, Bernoulli Naive Bayes, and a support vector machine (SVM) are used to classify texts in Persian.

2. MATERIALS AND METHODS

2.1. Methodology

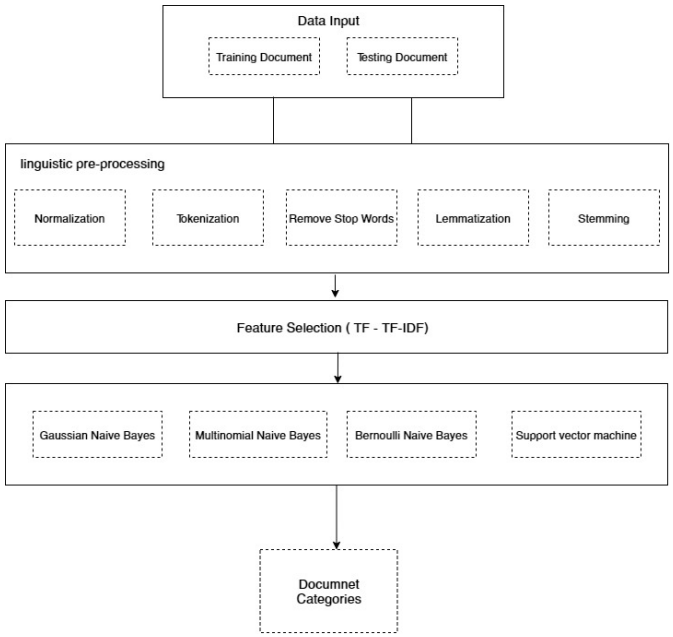

Fig. 1. The architecture of text categorization

The architecture of the work is divided into two parts: part of language model training and part of language model use.

2.1.1. Language model training

Input data are presented in text form, data are processed here, which consists of the following processes: normalization, tokenization, removal of stop words, lemmatization and stemming. This stage has a very strong impact on the results of text processing, as it is these processes that free language from specific linguistic features, which contributes to universalization of work with certain text. The next step is to convert text to a vector view using the TF IDF algorithm. In the last step, the word representation matrix obtained in the previous step is used as input. Here the language model is taught using each of the Bayes algorithms.

2.1.2. The language model used

As well as in training, the input is text that passes through all stages of linguistic processing. You then need to get a matrix representation of the words. To classify the text, we use the previously obtained language models and according to the standard metrics Precision and Recall we can compare the quality of these models.

We will discuss this architecture in more detail later in the article.

2.2. Text Pre-processing

Unlike other languages, including English, text mining for the Persian language has numerous difficulties due to the complexity of the linguistic structure and the type of written words. It is language features that prevent the creation of a universal natural language processing tool for all languages. We have identified a number of grammatical difficulties that a researcher may encounter in automatically processing

Persian text, and which are important to consider, as in the pre-processing phase. Let us examine in more detail some characteristic features that are characteristic directly of Persian grammar.

2.2.1. Relatively free order of words in Persian sentence

We deliberately emphasized the relativity of the free order of words, as by morphological typology Persian can be characterized as flective-analytic with agglutination elements. In comparison, with expensive languages, which refers to flective languages, where sentence members have the ability to freely occupy various places without changing their syntax roles, and which has a developed system of paddle endings, Persian looks more conservative, having a flective verb system with personal endings that vary in faces and numbers, but not in times. In turn, prominent modern and modal forms of the verb in Persian are expressed analytically. The main type of sentence in Persian is a two-part sentence having two principal members - subject and spoken. The function of the subject most often uses nouns and pronouns. In the function of the spoken is a verb. The structure of a simple common Persian sentence is as follows: subject (sometimes before it may be a circumstance of time) direct addition of indirect addition of the circumstance of place and mode of action said:

Daneshjoyan farda be Ordy Miravand. - The students are going on a tour tomorrow.
Haftye Ayandeh Be Moscow Parvaz Mikonam. – Next week I leave for Moscow.

Although this is a stable construction, it is nevertheless quite common in spoken speech to see deviations from this order. Compare the following offers:

In Kife Pol Ro Be Baradaram Bedeh – Give your brother this wallet.
In Kife Pol Ro Bedeh Baradaram – Give your brother that wallet.

In the latter version, the "normal" order of words is broken, which leads to the falling out of the pretext - Be.

In a situation of word order violation in the Persian sentence, the drop of the pretext is mandatory, otherwise it can become critical to the meaning content and lead to the loss of information:
In Kife Pol Ro Baradaram Bedeh - Give your brother this wallet. - such a proposal structure would most likely not be understood by the interviewer.

2.2.2. Individual differences in the writing of Persian words

At the present stage of development Persian language undergoes significant changes, it is possible to speak about its mobility and steady evolution. Although some traditions are characteristic of Persian, a certain part of spelling rules is assessed as outdated. They are replaced by updated trends in spelling, but to this day there are no clear spelling rules, and many words have significant differences in writing. For example, a number of pretexts, post-books, name affixes are written in some publications in a piecemeal manner, and in others separately. An example of a wobble in the fused or separate writing of verb affixes, in particular, may be an affix – **می** (mi). In modern spelling, separate spelling is accepted, the old spelling insists on ingot: (mishavad) **میشَوَد** and **می شَوَد**.
Different graphic expressions are also characteristic of words of Arabic origin or borrowed from European languages. This often applies to words that have sound [o]. In some options of writing it is designated by the Arab letter "vav" or only an "pish":
خاهر and **خواهر** - [khahar] ; **خاب** and **خواب**- [khab].

2.2.3. "Arabisms" in Persian

The problem of "Arabisms" in Persian has already been addressed in part by us in the previous paragraph. The main feature is the use of the Arabic alphabet on the letter, as it is Arabic that makes up the largest part of the Persian alphabet. It is in the interpretation of writing certain letters that most of the difficulties are concluded.

Traditional Iranian writing in isolated and finite positions of Arabic letter **ي** - ya - without dots results in that in those arabisms where is present **ى** - alif maqsurah ("limited A"), it is possible to mix these two letters. In modern writing standards the letter -ya is almost out of use, but by habit in many texts it is possible to find similar mistakes.

2.2.4. Defining the boundaries of a named group (isaphet construct)

A wide variety of syntax communication variants are found in Persian. The work [9] states that it is isaphet phrases that represent the most common type of phrase in Persian. Isaphet constructs are such a way of subordinating name phrases, in which the subordinate word in the first place is associated with a subsequent, subordinate word by means of an isaphet, i.e. a shock-free grammatical indicator - e (after the vowel - ye). This indicator in the grammar of Persian is referred to by grammarians as the "isaphet indicator". Isaphet links the definition and the word to be defined, for example:

لباس پدر [lebas-e man] - father's clothes

The structurally isaphet construction is as follows:

Defined + e + Definition

In this scheme, the unchangeable part is the definition following the defined word, which in turn gets an isaphet form by adding an isaphet indicator - e to the base.

2.2.5. Sense multiplicity (phenomenon of omonymism in Persian words and sentences)

The difficulties in parsing the sentence in Persian are largely related to the phenomenon of omonym. In some sentences, the syntax value of a sentence member is expressed by its place in the sentence, whereby changing the place results in changing their syntax function.

Madar farzand darad. The mother of the child has (the mother has a child).

Farzand madar darad. The mother has a child (the child has a mother).

Here, the order of the words in the Persian sentence is to distinguish between the syntax function to be and the complement. Similar problems may arise with the omonymic forms of the name parts of speech acting as the principal members of the proposal to be and spoken. Incorrect analysis of such constructs can be critical from a meaningful point of view.

2.2.6. The problem of omographs (caused, inter alia, by the lack of a way in Persian writing to express brief vowels "a," o, "e" in the middle of a word)

Continuing the consonant type of Arabic writing, Persian writing generally allows the recording of primarily consonants as well as long (in modern language stable) vowels using letters ا (alef) — â, و (vav) — and ی(ya) — i. For short (unstable) vowels, there is a system of Arabic harakats - superscripts of vowels. However, as in Arabic, publicity is used only in educational texts or rare cases where reading needs to be clarified. Words that differ only in short vowels do not differ on the letter.

For example, **کرم** [kerm] "worm", **کرم** [karam] "generosity", **کرم** [kerem] "cream", and **کرم** [krom] "chrome" register equally **کرم** [krm].

2.2.7. Features of word formation of plural nouns

In the singular, the noun in Persian is equal to the basis, two major agglutinative affixes are used to form the plural, always assuming emphasis:

ها-hâ — a virtually universal agglutinative affix attached to both nouns and non-nouns:

زن - [zan] – woman , **زنها** - [zanhâ] – women

ان-ân has position options **گان** –[gân] (after names on -e), **یان**–[yân] (after names on -â) and **وان**–[vân] (after names on -u) and joins mostly nouns denoting persons:

کارگر – [kârgar] – worker , **کارگران** - [kârgarân] – workers

خواننده – [xânande] – reader , **خوانندگان** - [xânandegân] - readers

Relatable it can also sometimes be used with some inanimate names:

چشم – [češm] – eye , **چشمان** - [češmân] – eyes

درخت – [daraxt] – tree , **درختان** - [daraxtân] - trees

In addition, Arabic methods of plural formation, borrowed together with Arabic vocabulary and also extended to some native words, are quite widely used:

ات-ât — affix of inanimate nouns

ین– **in**and **ون**-un — affixes of self-contained nouns that denote professions.

Arabic "broken" plural is a change of basis by replacing or adding vowels. It is a so-called "inner flexia," of which there are about forty models of formation. Such forms are usually always given in dictionaries. The plural is not used after numerals, in the meaning of gathering (when it comes to the subject at all) and in the name spoken:

دو روز – [do ruz] – two days

در باغ درخت زیاد است - [dar bâγ daraxt ziyâd ast] - there are many trees in the garden.

2.3.

This lack of proper tools for Persian language: There are not many tools for preprocessing and analyzing Persian text. For mining, documents must be preprocessed and information should be stored in an appropriate data structure for subsequent processing [10]. The purpose of preprocessing is to reduce the space dimensions of the terms in the document that usually includes the following steps:

2.3.1. Removing the stopword

Despite being repeated in most texts, these words are not meaningful and lack conceptual information. In text processing, it is necessary to find the words that help to achieve a better model. In this study, in addition to common stopwords, the demonstrative, pronominal-adjective, conjunctions, group conjunctions, reflexive pronouns, emphatic pronouns, interrogative pronouns, numbers, numbers in letters, postpositions, auxiliary verbs and copulas are collected that make up 940 stopwords.

2.3.2. Text normalization

One of the preprocessing stages of text data in the field of text mining is the normalization and harmonization of texts. Before analyzing any data, noises and data must be isolated from other data, which will increase the accuracy of analysis and reduce the amount of processing. Removing punctuation marks, non-Persian words, spaces between words, abbreviations, and word rooting are among the most important actions at this stage.

Weighting words:

One of the important issues in text processing is to examine the role of words and their influence on the text. Using different weighting patterns for each word at this stage, the influence of a word is compared with other words used in the text. The reason for this is to facilitate the meaning comprehension of the concept of a document with a proper weighing criterion. In this research, a TF and TF-IDF based common method is used in Bayesian algorithms and Word embedding is devised in vector algorithms that the performance of each one is described below.

2.3.3. Tokenization

This part must have the ability of sentence recognition in the input text regarding the sentence divider characters in the Persian language. To create this device, first all symbols, characters, especially syntactic rules which break the sentences, must be identified. Since the sentence is necessary for many language processes, the accurate outcome of this section is of high importance [11].

2.3.4. Lemmatization

Usually, texts contain different grammatical forms of the same word, and one-root words may also occur. Their goal is to bring all the word forms encountered to a single, standard vocabulary form.

Lemmatization is a more subtle process that uses vocabulary and morphological analysis to ultimately bring the word to its canonical form(lemma) [12].

2.3.5. Stemming

Stemming is to find out the roots of words by eliminating their prefixes and suffixes in a way that words with similar root become an identical form.

The difference is that the stemmer operates without knowing the context and, accordingly, does not understand the difference between words that have different meanings depending on the part of speech. However, the Stemmers have their advantages. The most common purpose of rooting is to homogenize words and verbs written in different grammatical forms. This can increase the effectiveness of the system [13].

2.4. Feature weighting methods

Most of the algorithms in machine learning cannot process strings or plain text in their raw form. Instead, they require numbers as inputs to be able to function. By transforming words into vectors, word embeddings, therefore, allows us to process the huge amount of text data and make them fit for machine learning algorithms.

Word embeddings are the process by which words are transformed into vectors of real numbers. I this article, words were modeled using TF-IDF representation for transforming into vectors used in the Bayesian algorithms.

2.4.1. TF based approach

In this simple and very practical method, which was first proposed, in the case of existence of T_k feature in the document d_i, its weight will be equal to the number of feature repetitions in the corresponding document.

$$w_{ki} = tf(t_k, d_i) = \begin{cases} (t_k, d_i) & t_k \in vector\ of\ d_i \\ 0 & t_k \notin vector\ of\ d_i \end{cases} \quad (1)$$

Where, (t_k, d_i) is equal to the number of replicates of each feature t_k in the document d_i.

2.4.2. IDF-based method

In these methods, weighting the features is a function of the distribution of t_k feature distribution within the D documents. The main idea of weighing in this category is that as the number of documents with

the $t_k$features decreases, t_k is a more appropriate feature to distinguish the documents from one another and should have more weight. This method is used in the field of data retrieval for the first time.

$$w_{ki} = idf(t_k, d_i) = \log\left(\frac{|D|}{|D(t_k)|}\right) \quad (2)$$

Where, |D| is the total number of documents and $|D(t_k)|$ is the number of documents from the set D that t_k features exist in them.

2.4.3. TF-IDF method

This technique, which is one of the most common weighting methods for this category, is the combination of TF and IDF-based methods [14,15].

In this method, the repetition rate of a word in a document vs. its replication in the set of all documents is considered. The weight of the words is determined by term frequency and inverse document frequency criteria and finally the weight of each word is calculated through the following formula.

$$w_{ki} = TFIDF(t_k, d_i) = tf(t_k, d_i) * idf(t_k, d_i) \quad (3)$$

2.5. Classification

Data mining and machine learning algorithms that perform classification accept a matrix as an input and learn the pattern of each class from this matrix and its features. Then, if a new sample with unknown class is given to the trained algorithm, this algorithm can classify this new instance to the possibly right class. In general, the algorithms that have right classes during their training in their matrix are called supervised learners. The following section addresses the Bayesian algorithm and the support vector machine (SVM) method [16-20].

2.5.1. Bayesian method

This method is one of the most widely used methods in classifying texts. In this approach, the text is considered as sets of words independent of each other and independent of its location in the text. The probability function of each text is derived from the product of the probability of its words and the probability of a text occurrence with that is length. System learns with estimating the parameters for generating a model that only uses tagged texts. The algorithm uses the estimated parameters to classify new texts by calculating which categories have the most similarity to the given text [21,22,23].

The class of a document corresponds to words that appear in a document that the following formula is used to estimate the document's class:

$$P(y\,|x_1, \dots, x_n) = \frac{P(y)P(x_1,\dots,x_n\,|\,y)}{P(x_1,\dots,x_n)} \quad (4)$$

It is assumed that the values of attributes with the values of the objective function are conditionally independent of each other. This assumption implies that under the condition of viewing the output of the objective function, the probability of observing $x_1, \dots, x_n$ will be equal to the product of each attribute separately.

$$P(x_i\,|y, x_1, \dots., x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i\,|\,y) \quad (5)$$

$$P(y\,|\,x_1, \dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i\,|\,y)}{P(x_1,\dots,x_n)} \quad (6)$$

Given that the denominator of this fraction does not depend on the type of class, it is possible to consider the denominator constant and in accordance with the joint probability distribution rule, the formula can be written as follows:

$$P(y\,|x_1, \dots, x_n) \propto P(y)\ \prod_{i=1}^{n} P(x_i\,|\,y) \quad (7)$$

The above steps are model training. In the next step, the estimated probabilities to determine the new sample class could be calculated as the following formula:

$$\hat{y} = argmax\ P(y)\prod_{i=1}^{n} P(x_i\,|\,y) \quad (8)$$

Three popular Naïve Bayes algorithms:

2.5.2. Gaussian Naïve Bayes

When attribute values are continuous, an assumption is made that the values associated with each class are distributed according to Gaussian i.e., Normal Distribution. The likelihood of the features is assumed to be Gaussian:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \quad (9)$$

The parameters σ_y and μ_y are estimated using maximum likelihood.
Maximum likelihood Estimates:
Mean:

$$\hat{\mu}MLE = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (10)$$

Variance:

$$\hat{\sigma}^2\ MLE = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu})^2 \quad (11)$$

Gaussian good for making predictions from normally distributed features

2.5.3. Multinomial Naïve Bayes

Multinomial naive Bayes implements the naïve Bayes algorithm for multinomially distributed data, and is one of the two classic naïve Bayes variants used in text classification. With a multinomial event model , feature vector represent the frequencies as word vector counts have been generated by a multinomial, the distribution is parametrized by vectors for each class $\theta_y = (\theta_{y1}, \ldots, \theta_{yn})$Y,where n is size of the vocabulary [24,25].
$\widehat{\theta_{yi}}$ is estimated by a smoothed version of Maximum likehood:

$$\widehat{\theta_{yi}} = \frac{N_{yi} + \alpha}{N_y + \alpha n} \quad (12)$$

Where $N_{yi} = \sum_{x \in T} x_i$ is the number of times feature i appears in sample of class y in the training set T
$N_y = \sum_{i=1}^{n}$ is the total count of all features for class y

The smoothing priors $\alpha \geq 0$ accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting $\alpha = 1$ is called Laplace smoothing, while $\alpha < 1$ is called Lidstone smoothing [26].
Multinomial good for when your features (categorical or continuous) describe discrete frequency count.

2.5.4. Bernoulli Naïve Bayes

Bernoulli Naive Bayes is used on the data that is distributed according to multivariate Bernoulli distributions. i.e., multiple features can be there, but each one is assumed to be a binary-valued (Bernoulli, Boolean) variable. So, it requires features to be binary valued.

$$p(x \mid \mu) = \mu^x(1 - \mu)^{1-x} = exp\left(x \ln\frac{\mu}{1-\mu} + \ln(1 - \mu)\right) \quad (13)$$

Bernoulli good for making predictions from binary features

2.5.5. Performance Measures

Many measures are used to evaluate various aspects of text processing and information retrieval system. The performance of such a system, which is designed to classify document to their categories, is often gauged in terms of precision, recall and macro-average [27]. Let Tue Positives (TP) be the number of documents that are classified as relevant, judged by the human and the classifier Tue Positives , False Negatives (FN) be the number of documents that are classified as relevant by judgment of the human and irrelevant by judgment of the Classifier FN, False Positives (FP) be the number of documents that are classified as irrelevant by judgment of the human and relevant by judgment of the classifier FP and True Negatives (TN) be the number of documents that are classified as irrelevant by judgment of the human and the classifier TN. Recall and precision are defined respectively as:

- Precision: Measures that have a high ability to retrieve document that are judged by the user as being relevant

$$Precision = \frac{TP}{TP + FP}$$

- Recall: Measures that have a high ability of the search to find all of the relevant item in the corpus.

$$Recall = \frac{TP}{TP + FN}$$

2.5.6. Experimental Results

The dataset used in our system consists of 10000 documents, distributed into ten categories: Law, Religion, Economic, Health, Scientific, Political, Educational, Cultural, Social, Sport have been collected from Irna News Website. The dataset was collected from the Irna website and divided into a 75 % document training set and a 25% document testing set

3. RESULTS

Since this paper deals with three objectives of comparing the weighing methods, the effect of reducing the feature vector and selecting the best machine-learning algorithm for classifying Persian texts, in this section, 4 tests are done on the texts. In the first method, using the TF weighing method and the removal of the prepositions, it is attempted to compare the algorithms that the Multinomial algorithm with the accuracy of approximately 83% has the best result compared to the rest of the algorithms. In the second method, the algorithms are compared without eliminating the prepositions that the above algorithm outperforms the rest of the algorithms by 82.4% accuracy but it is less than the first method and the duration of the training is increased due to the large feature vector.

Table 1. TF method without stopwords

#	Algorithm	Precision (Micro)	Precision (Macro)	Recall (Micro)	Recall (Macro)	F1 Score (Micro)	F1 Score (Macro)
1	MultinomialNB	0.828000	0.834406	0.828000	0.829661	0.828000	0.822319
2	GaussianNB	0.580000	0.583903	0.580000	0.581826	0.580000	0.577923
3	BernoulliNB	0.803600	0.801278	0.803600	0.805042	0.803600	0.798569
4	SVM	0.769600	0.767363	0.769600	0.771154	0.769600	0.768506

Table 2. TF method with stopwords

#	Algorithm	Precision (Micro)	Precision (Macro)	Recall (Micro)	Recall (Macro)	F1 Score (Micro)	F1 Score (Micro)
1	MultinomialNB	0.824800	0.836364	0.824800	0.826687	0.824800	0.817535
2	GaussianNB	0.590400	0.598527	0.590400	0.591851	0.590400	0.590216
3	BernoulliNB	0.802000	0.799499	0.802000	0.803220	0.802000	0.797958
4	SVM	0.759600	0.758481	0.759600	0.761983	0.759600	0.758634

Table 3. TF-IDF method without stopwords

#	Algorithm	Precision (Micro)	Precision (Macro)	Recall (Micro)	Recall (Macro)	F1 Score (Micro)	F1 Score (Micro)
1	MultinomialNB	0.842800	0.844971	0.842800	0.844175	0.842800	0.838530
2	GaussianNB	0.604400	0.606828	0.604400	0.605730	0.604400	0.603463
3	BernoulliNB	0.818000	0.814981	0.818000	0.819134	0.818000	0.813763

Table 4. TF-IDF method with stopwords

#	Algorithm	Precision (Micro)	Precision (Macro)	Recall (Micro)	Recall (Macro)	F1 Score (Micro)	F1 Score (Macro)
1	MultinomialNB	0.834000	0.836473	0.834000	0.835223	0.834000	0.829738
2	GaussianNB	0.592400	0.591833	0.592400	0.593921	0.592400	0.590132
3	BernoulliNB	0.818800	0.818432	0.818800	0.819831	0.818800	0.815108

Then the TF-IDF method is used to calculate the features of the words with and without prepositions to compare the efficiency of the algorithms [28,29]. The Multinomial algorithm has a good accuracy of 84.3% with the removal of prepositions. Removing the prepositions significantly helps to reduce the feature vector, which results in increased processing speed and accuracy, as shown in Figure 2.

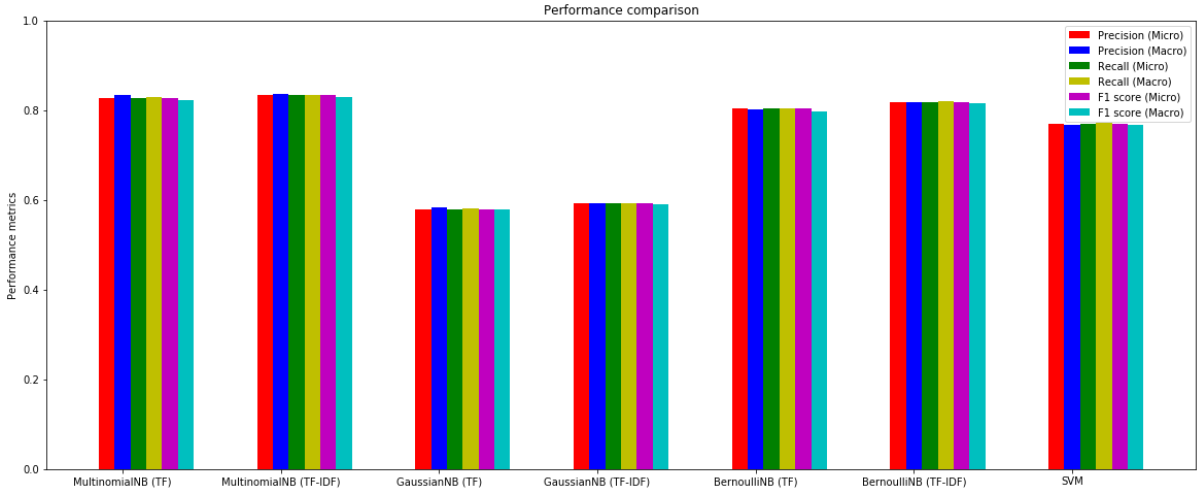

Fig. 2. Precision, Recall and F1 score for all algorithms

There's a reason why the two classic variants of NB on text classification are Bernoulli and Multinomial: these are discrete distributions. Gaussian, on the other hand, is a continuous distribution. Basically, to use this with text classification, you have to figure out a way to represent a word as if it were drawn from a normal distribution. A Gaussian naive bayes classifier is used with a continuous stream of text data. This allows an algorithm to begin classification before the entire block of new text is available. The basic idea is that given the currently known text the probability of expected words can be used to help a user complete a sentence or classify immediately with an educated guess [30].

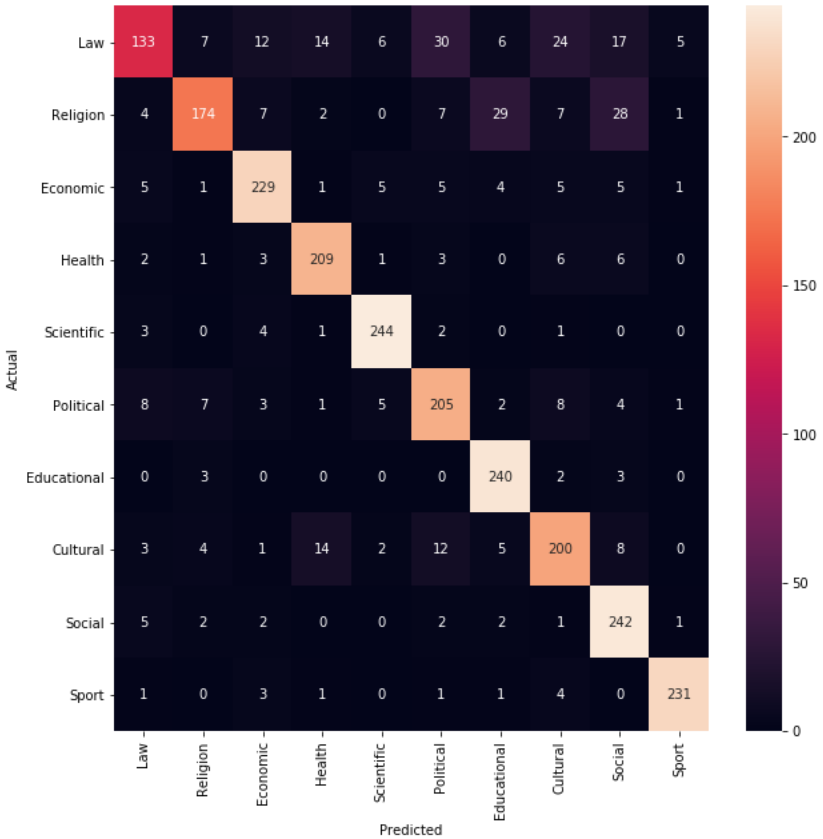

Fig. 3. Confuse Matrix for best result (Multinomial algorithm)

4. DISCUSSION AND CONCLUSION

Due to the increasing volume of access to textual sources, the automatic classification of text data is an important issue. This paper presents a method for classifying Persian texts. A general method based on machine learning contains the learning and testing phases. In general, the text classification process has three main parts of preprocessing, feature selection, and learning algorithm. In the feature selection section, by keeping the useful meaning of the words, eliminating noise and words that have little information load, they can reduce the dimensions of the feature vector to some extent and prepare them for the next steps. Based on the results of the preprocessing and feature selection, the TF-IDF method has a direct impact on the accuracy and speed of the learning algorithm. The classification is not only used to find the subject of the text, but also applied in the filtering of the texts with respect to their relative content.

The problem of overcoming language linguistic specificity is one of the most complex in word processing processes. We have previously formulated problems related to the processing of Persian, such as, relatively free order of words, differences in the writing of Persian words, "arabisms," the presence of isaphetic construction, sense multiplicity, the problem of omographs, the peculiarities of word formation of the plural number of nouns. All these tasks are solved by means of the stage of linguistic text processing, which allows to interact with natural language as with a more universal design. The algorithms we used to classify text don 't work with words, but only vector data. In the course of language model training, it is necessary to work with big data, and vector representation of words takes a lot of RAM. Stop words generally occupy most of the text and do not bear a serious informative meaning. As a result of our research, it can be seen that due to the removal of stop words from the text, the volume of vector transformation decreases by 25%. In our study, we compared different Bayes network algorithms such as, Gaussian NB used when all our features are continuous. We can't represent features in terms of their occurrences. Bernoulli NB. It assumes that all our features are binary such that they take only two values. Means 0 can represent «word does not occur in the document» and 1 as «word occurs in the document". Multinomial NB. It is used when we have discrete data. In text learning we have the count of each word to predict the class. Words is represented in terms of their occurrences. In the process of comparing said algorithms, the best results were shown by the Multinomial NB algorithm. When working with it, the accuracy of text classification with stop words was 83%, and the accuracy of text classification without stop words - 82.4%, which gives almost identical results of classification. We recommend for future studies is to improve different TF-IDF smoothing techniques to improve the results of classification algorithms.

ACKNOWLEDGEMENTS

This publication has been prepared with the support of the "RUDN University Program 5-100".

REFERENCES

[1] Hinnebusch, M., Darnashek, M., and Cohen, J. Visualizing document classification: A search Aid for the digital library. IBM T.J Watson Research Center.
[2] T. Pilehvar, H. Faili, M. Soltani, Classification of Persian textual documents using Learning Vector Quantization,4rd IEEE Conference on Knowledge Engineering and Natural Language Processing,NLP-KE,2009
[3] Carl Benedikt Frey and Michael A. Osborne "THE FUTURE OF EMPLOYMENT: HOW SUSCEPTIBLE ARE JOBS TO COMPUTERISATION",September 17, 2013
[4] Ko Y., "Text Categorization using Unlabelled Data," PhD Dissertation University Seoul, Korea,2003.
[5] K. G. Al-Shalabi R., Gharaibeh M., " Arabic Text Categorization Using kNN Algorithm," presented at the Proceedings of the Int. multi conf. on computer science and information technology, 2006.
[6] Alhutaish, Roiss & Omar, Nazlia. (2015). Arabic Text Classification Using K-Nearest Neighbour Algorithm. International Arab Journal of Information Technology. 12. 190-195.
[7] Saleeb H., "Information Retrieval: A Framework for Recommending Text-Based Classification Algorithms," PhD Doctor of Professional Studies, Pace University, 2002.
[8] Xu, Shuo & Ma, Fujing & Tao, Lan. (2007). Learn from the Information Contained in the False Splice Sites as well as in the True Splice Sites using SVM. International Journal of Computational Intelligence Systems. 10.2991/iske.2007.13.
[9] Hasan Zadeh P, Valipour A. On the issue of isafet constructions in the Persian language // Young scientist. — 2014. — no15. — C. 410-412
[10] Rezaeian, N. and Novikova, G.M. Morphological and syntactic analysis of persian text with conditional random fields. International research journal, 2016
[11] for Automatic Indexing // Commun. ACM. — 1975. Vol. 18, no. 11. — Pp. 613–620.
[12] T Müller, R Cotterell, A Fraser, H Schütze Joint lemmatization and morphological tagging with lemming. In EMNLP-2015,pp.2268-2274
[13] Amir Azim Sharifloo , Mehrnoush Shamsfard, A Bottom up Approach to Persian Stemming, Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-{II}
[14] Albitar, Shereen & Fournier, Sébastien & Espinasse, Bernard. (2014). An Effective TF/IDF-based Text-to-Text Semantic Similarity Measure for Text Classification. 10.1007/978-3-319-11749-2_8.

[15] Salton, G., Yang, C.S., “On the Specification of Term Values in Automatic Indexing”, Journal of Documentation, Vol. 29, No. 4, pp. 351-357, 1973.
[16] M. Farhoodi, A., Yari, M. Mahmoudi., "A Persian Web Page Classifier Applying a Combination of Content-Based and Context-Based Features_", International Journal of Information Studies
[17] M. Farhoodi, A., Yari, M. Mahmoudi., "A Persian Web Page Classifier Applying a Combination of Content-Based and Context-Based Features_", International Journal of Information Studies
[18] Xu, Shuo & Ma, Fujing & Tao, Lan. (2007). Learn from the Information Contained in the False Splice Sites as well as in the True Splice Sites using SVM. International Journal of Computational Intelligence Systems. 10.2991/iske.2007.13.
[19] Alhutaish, Roiss & Omar, Nazlia. (2015). Arabic Text Classification Using K-Nearest Neighbour Algorithm. International Arab Journal of Information Technology. 12. 190-195.
[20] Ahmadi, Parvin & Tabandeh, Mahmoud & Gholampour, Iman. (2016). Persian text classification based on topic models. 86-91. 10.1109/IranianCEE.2016.7585495.
[21] H. Zhang (2004). The optimality of Naive Bayes. Proc. FLAIRS.
[22] Xu, S. (2016). Bayesian Naïve Bayes classifiers to text classification. Journal of Information Science, 0165551516677946.
[23] McCallum A. et al. A comparison of event models for naive bayes text classification //AAAI-98 workshop on learning for text categorization. – 1998. – Т. 752. – №. 1. – С. 41-48.
[24] Ting, S. L., Ip, W. H., & Tsang, A. H. (2011). Is Naive Bayes a good classifier for document classification. International Journal of Software Engineering and Its Applications, 5(3), 37-46.
[25] He, F., & Ding, X. (2007, April). Improving naive bayes text classifier using smoothing methods. In European Conference on Information Retrieval (pp. 703-707). Springer Berlin Heidelberg.
[26] Kim S. B. et al. Some effective techniques for naive bayes text classification //IEEE transactions on knowledge and data engineering. – 2006. – Т. 18. – №. 11. – С. 1457-1466.
[27] N. Rezaeian, G.M. Novikova,Detecting Near-duplicates in Russian Documents through Using Fingerprint Algorithm Simhash,Procedia Computer Science,Volume 103,2017
[28] F. Peng and D. Schuurmans, Combining naive Bayes and n-gram language models for text classification, in Lecture Notes in Computer Science, (2003) 335-350.
[29] W. B. Cavnar and J. M. Trenkle, N-gram-based text categorization, in 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR-94) (1994).
[30] Ahmed H. Aliwy and Esraa H. Abdul Ameer, Comparative Study of Five Text Classification Algorithms with their Improvements, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 14 (2017) pp. 4309-4319