Development of text classification system

A Sabyrkulova*

Faculty of Information Technology, Kazakh-British Technical University, Tole bi Str.59, Almaty, Kazakhstan

**Corresponding author's e-mail: sabyraliya@gmail.com*

Abstract

In this article we will consider the approaches applied to classification of texts in a natural language processing by their topics. The problem of automatic classification of texts is considered by sentiment analysis, there are described methods of machine learning for the solution of this problem. The description of an algorithm of classification of the naive Bayesian classifier is provided. Utilization of classifiers, allows to limit search of necessary information to rather small subset of documents.

Keywords: bag of words, machine learning, Bayesian classifier

1 Introduction

The Internet is full of information and every day we deal with constantly increasing volume of the processed and accumulated information that does a problem of text classification more and more urgent. For example, such necessity at automatic processing of a news stream and distribution of news texts by catalogs, Email Classification and Spam Filtering, recognition of emotional coloring of texts in documents. In all cases, the main idea is to assign the suitable category or label to each document that needs to be classified.

2 Goals

The main purpose of this work is creation of an automatic texts classification system. The idea is to build web-based platform by using well-known NLP tools and libraries for the reading and processing widely distributed and dynamic data for classification into different categories.For the construction of a system it is necessary to know the Naive Bayesian Classifier, sentiment analysis, moreover system will build on high-level web framework Django, and written in python language with different useful library such as OpenNLP, NLTK.

3 Feature Representation

Document representation and feature selection,it's a one of the most fundamental tasks which needs to be realized, preceding any classification problem.To understand better we use 'bag-of-words' (bag-of-features or bag-of-keypoints) [1] word-based document representation method, where we ignore the structure and sequence of words in the document. It is regularly used in methods of text classification where the frequency of each word is used as a feature for training a classifier. The feature vectors describe the words noticed in the documents. By this representation a document is considered to be simply a group of words which occur in it at least once. With this technique, it is possible to have thousands of words occurring in a small set of texts.

4 Classification Algorithm. Naive Bayes classifier

Nowadays we have different well-known often used machine learning algorithms for text classification such as decision trees, support - vector machines, naïve Bayes classifier, nearest neighbor algorithms.

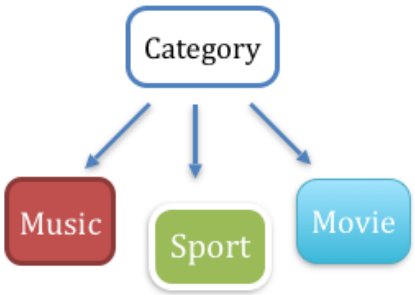

FIGURE 1 Classification by category

The naive Bayesian classifier [4] (NBC) is one of examples of use of methods of the vector analysis. This model of classification is based on a concept of conditional probability of the document d to a class c. NBC – one of the most often used classifiers, because of comparative simplicity in implementation and testing.

Basic formula of the naive Bayesian classifier (1):

$$P(A\backslash B) = \frac{P(B\backslash A\_P(A)}{P(B)}. \quad (1)$$

For this model, the document is a vector: d = { w1, w…, wn }, where wi - weight of i-th term, and n – size of the dictionary of selection. Thus, according to Bayes's theorem, probability of a **c** class for the document **d** will be (2):

$$P(c\backslash d) = \frac{P(d\backslash c)P(c)}{P(d)}. \quad (2)$$

Thus, for finding of the most probable class for the document d = { w1, w…, wn }by using naive Bayesian classifier, it is necessary to count conditional probabilities of accessory of the document d to each of the provided

classes separately and to choose the class having the maximum probability:

$$c_{NB} = argmax_c[P(c_j) * \prod_i P(\omega_i \backslash c_j)]. \quad (3)$$

```
import nltk
from nltk import word_tokenize, WordNetLemmatizer
from nltk.corpus import stopwords
from nltk import NaiveBayesClassifier, classify
from collections import Counter
import os
import random
import time
import io


lemmatizer = WordNetLemmatizer()

def getDataAsList(folder):
    out_lst = []
    file_list = os.listdir(folder)

    for fl in file_list:

        try:
            curr_file = io.open(folder + fl, "r", encoding="utf8")
            out_lst.append(curr_file.read())
        except UnicodeDecodeError:
            pass

    return out_lst


def preprocess(sentence):
def getFeatures(text, model=""):
def train(features, samples_proportion):
def evaluate_performance(train_set, test_set, classifier):
if __name__ == "__main__":
    stopwords_list = stopwords.words("english")

    spam_emails = getDataAsList("data/aliya/spam/")
    spam_emails_len = len(spam_emails)

    ham_emails = getDataAsList("data/aliya/ham/")
    ham_emails_len = len(ham_emails)


    all_emails = [(email, "spam") for email in spam_emails] + [(email, "ham") for email in ham_emails]

    assert(len(all_emails) == ham_emails_len + spam_emails_len)


    random.shuffle(all_emails)
```

There is a small problem connected to this formula. If a word never appears in the given training data, its relative frequency estimate will be zero.For the solution of this problem we applied the Laplace law of succession [2] to estimate P(wi|c j). The estimate of the probability P(wi|cj) is given as:

$$P(\omega_i \backslash c_j) = \frac{n_{ij}+1}{n_j+k_j}, \quad (4)$$

where nj is the total number of words in class cj , nij is the number of occurrences of word wi in class cj and kj is the vocabulary size of class cj . This is the result of the Bayesian estimation with a uniform prior assumption [3] Bayesian method has the high speed of work and simplicity of mathematical models.This method is often used as a basic method when comparing various methods of machine learning.

5 Example

In order to better understand how naive Bayesian classifier works, let's assume that we are faced with a spam problem,and we need to create simple Spam filter which will attempt to classify incoming email messages in two different categories as 'spam', 'ham' (good, non-spam email) or 'unsure'. To solve this problem we will manually create data set with «emails», and determine bag of words which usually come in spam such as "sale", "buy now", "lowest prices", "click here» and etc. By counting probability of how many times spam words appears in email, system will identify, is it spam or ham. The program is written in Python with using NLTK libraries.

In result, we retrieve information about email in the table:

email	class
Sales! come and take your favorite boots	spam
Hi Mariya! Whats going on?	ham
Your application was declined. Please come tomorrow	ham
Viagra is available in our site. Order now	spam
Sale Sale Sale	spam
Buy now! only one day lowest prices	spam
Can you bring my book Mariya	ham
Hey hi! i'm going to celebrate my birthday,please come	ham

6 Conclusion

Nowadays the classification problem is one of the most fundamental problems in the machine learning and data mining. The amount of data that needs to be processed is very large, consequently, text analysis techniques must be designed effectively to manage large numbers of elements with varying frequencies. In this research paper we mentioned work with classification algorithms, moreover have got acquainted with the principle of Bayesian classifier work and reviewed an example of simple spam filter where we categorized e-mails in two groups (spam and not spam).

References

[1] Barakhnin V, Lukpanova L, Solovyov A 2014 An algorithm for constructing word forms using inflected classes for systems of morphological analysis of the Kazakh language *Bulletin of NGU, Series: Information technology* **12**(2) 25-31 (*in Russian*)
[2] Belonogov G, Zelenkov Y 1985 Algorithm for automatic analysis of Russian *word Questions of information theory and practice* **53** 62-93 (*in Russian*)
[3] Valiayeva T *The grammar of the Kazakh language* http://kaz-tili.kz (*in Russian*)
[4] Fedotov A, Tusupov D, Sambetbayeva M, Yerimbetova A, Bakiyeva A, Idrisova I 2015 The model determine the normal form of the word for the Kazakh language *Bulletin of NGU, Series: Information technology* **13**(1) 107-16 (*in Russian*)
[5] Porter M F 1980 *An algorithm for suffix stripping Program* **14**(3) 130–7
[6] Bakiyeva A *Program generation of word forms of the Kazakh language* http://poem.ict.nsc.ru/~bakieva_aigerim/kazGen/ (*in Russian*)