National Research University Higher School of Economics
Faculty of Computer Science HSE and University of London Double Degree
Programme in Data Science and Business Analytics

# TERM PAPER

## Classification and routing of incoming correspondence with specific vocabularies

Prepared by student of Group 182 in Year 3 Yakunina Kseniia

Term Paper Supervisor: A.V. Helvas

Moscow, 2021

**Abstract**

The main goal of the project is to create a training sample for a thematic area, containing a large amount of special terminology, and to create a solution for the classification and routing of official documents with the parallel use of formal algorithms and neural network analysis.

# Contents

**Glossary**

**Classification problem** — the problem of dividing a given set of *objects* into some *classes*.

**Feature** — a characteristic of an object, such that for objects belonging to the same class, the values of these characteristics have similar values.

**Classifier** — an algorithm that solves the classification problem (takes an object as input and assigns it to a certain class).

**Binary Classifier** — Algorithm for solving classification problem with two classes.

**Train the classifier** — an algorithm for finding the optimal parameters for the classifier.

**Training sample** — a set of objects used to train the classifier. For each class of the considered classification problem, there are objects from this set.

**Accuracy** — quality metric, the proportion of correct responses of the classifier on a certain sample.

**Embedding** — a method of converting text to vector or matrix form for further feeding to the neural network input.

**N-grams** — of a sequence of N consecutive elements

# 1 Introduction

## 1.1 Work structure

The first section of the work contains the statement of the problem

The second section formalizes the problem and describes the mathematical models used to build algorithms.

The third section of the work provides an overview of the scientific and technical literature on the research topic.

The fourth section describes an approach to solving the problem.

In the fifth section, descriptions of the experiments carried out in the course of the work are given.

## 1.2 Work Description

This work represents a project for the clustering of text documents by keywords in Russian, in particular, patent abstracts. This is a clustering task. Algorithm testing is carried out in two stages. First, the results are analyzed on the trained sample, and then the clustering analysis is used for a new sample.

## 1.3 Relevance of work

For any company, the process of sorting the incoming flow of documents can be costly in terms of time and human resources. Therefore, it is much more profitable for companies to use automatic sorting of documents, for example, to immediately send a document to the necessary department for work. This will significantly reduce the cost of manual processing of documents and allow more efficient allocation of resources.

## 1.4 Goals and objectives

The aim of the work is to create a training sample for a thematic area containing a large amount of special terminology and to create a solution for the classification and routing of official documents with the simultaneous use of formal algorithms and neural network analysis.

To achieve this goal, the following tasks were solved in the work:

**Objectives:**

— Review of methods for solving similar problems
— Formation of a database consisting of patent texts.
— Data prepossessing and analysis
— Clustering product for patents
— Analysis and comparison of the results obtained

## 2 Formulation of the problem

As input data there are abstracts of Russian patents in text format. It is necessary to build a model that would predict which area each patent belongs to by keywords, try to determine which areas the algorithm was able to highlight, and also determine which words are significant for analysis. It is also necessary to test the algorithm on a new sample and compare the results.

## 3 Formal statement of the problem

Let $\boldsymbol{X}-$ be a a set of text objects

$$(x_1, \ldots, x_i)$$

and $\boldsymbol{Y}-$ be a set of clusters

$$(y_1, \ldots, y_i)$$

There is a finite sample of texts

$$X^n \subset X$$

which needs to be split into subsets of clusters Y, so that for any

$$x_j \in X^n$$

a cluster has been defined

$$y_k \in Y$$

, objects of different clusters should have significant differences.

# 4 Literature review

There are a lot of methods for analyzing and processing natural language and applying approaches in various fields.

For example, there is a project Natasha parser [1] - it is a set of libraries for natural language processing, with the help of it you can perform morphological and semantic analysis of texts in Russian language.

To assess the similarity of words in meaning, there is a solution Word2vec [3], which translates words into vector format and with the power of this you can determine whether the words are synonyms, this approach also helps to perform cluster analysis of words.

Project Bidirectional Encoder Representations from Transformers (BERT)[4] -is a converter-based machine learning technique for natural language processing (NLP) pre-learning developed by Google. This model was trained on a large number of texts, this greatly simplifies the development of a new classifier, since there is no need to build a model from scratch.

In the paper "The method of frequency-morphological classification of texts" [5] describes the use of frequency and morphological analysis to solve the problems of text classification. In this paper, classification is considered using decision trees, as well as using combined methods.

In the study book "Introduction to Information Retrieval" [2] in chapter 13 presents mathematical approaches to the classification of texts, such as the Naive Bayes approach and The Bernoulli approach.

There is a ready-made algorithm called IPCCAT [12] that helps to determine which class of a patent belongs to by keywords.

# 5 Solution Description

## 5.1 Data collection

Patent data in Russian was downloaded using the "google patents" service [9], a dataset with patent information for the entire 2020 was downloaded from this site. Further, the dataset also contained information about the name of the patent, its authors, as well as a link to the web document of the patent.

With the help of this dataset, the patent data in PDF format was downloaded and then converted into text format. The documentation of patent applications has a clear structure, so it is easy enough to extract the necessary information from this data. To train the model, only the abstract part of the patent and the name of the patent are needed, therefore, only these fragments were extracted from the entire text using regular expressions.

The text data has been translated into a "DataFrame" with the structure described in the table 5.1

Table 5.1 — Data Frame

| Filename | Full Text | Abstract |
|---|---|---|

Since texts are a large amount of information and their processing requires a lot of computing power, and even processing 25,000 patents (all patent applications for 2020) is problematic, therefore, a training sample of 2,000 patent texts was formed for training, and a test sample was also prepared from 2000 texts of patents. The next step was to prepare data for training.

## 5.2 Data pre-processing

First of all, the data needs to be cleared of punctuation symbols and stop words. This stage was performed using the ntlk [6] library and a ready-made set of stop words for the Russian language and punctuation symbols. It is important to clear the data of stop words, since they do not carry a semantic load, so they will only interfere with the training of the model.

The next stage of data pre-processing is the product of data tokenization and lemantization. To process a large text block, it is better to divide it into small segments, in our case, into separate words. For this, the tokenization method is used. Tokenization was done using the ntlk [6] library. Lemantization is used to bring words to normal form, which ultimately allows for better frequency and morphological analysis, as well as improving the quality of the model. For lemantization, libraries from standford nlp [10] with pre-trained models in Russian were used.

In addition to this, data with missing values was removed from the set date, that is, those data from which it was not possible to correctly extract word tokens.

An important stage in the pre-processing of texts is the extraction of statistical indicators; this can be done using frequency and morphological analysis. This is important for understanding which words are key for a particular text, and which are often found in all patent texts. Typically, keywords are words with a frequency of no more than 20%. As a result of the completion of data processing, a "DataFrame" was formed with the structure described in the table 5.2:

Table 5.2 — Data Frame

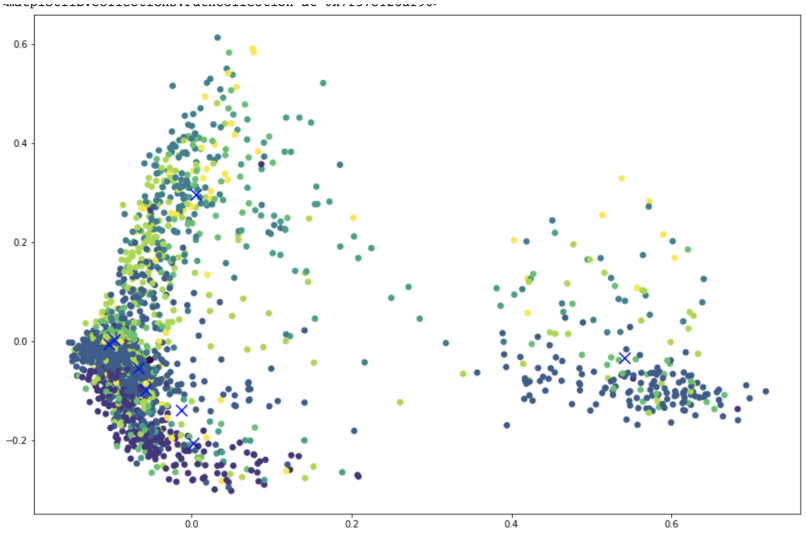
| Filename | Full Text | Abstract | Tokens |
|---|---|---|---|

### 5.3 Cluster Analysis

Before clustering texts, you need to encode them using the Tf-idf algorithm, it converts words into vector format and extracts meaningful words so that there is less noise in the data. In this work, we used the built-in algorithm of the scikit-learn [11] library called TfidfVectorizer.

In Figure 5.1, there is a noticeable difference in the quality of clustering using the Tf-idf algorithm. Data with encoding by Tf-idf algorithm show better performance and clusters are visibly distinguishable, that is why it is beneficial to use it in the work.

Before starting the cluster analysis, it is necessary to determine the optimal number of clusters for our data.



Clastering without TD-IDF

Clastering with TD-IDF

Figure 5.1 — Clastering and TD-IDF

For this, the "Elbow Method" [7] was used in the work. After using this method, the optimal number of clusters = 8 was established.

For clustering, the two methods Kmeans and Kmeans will be used together with a pre-trained model based on Bert[4].

# 6 Experiment

## 6.1 Model Implementation

An algorithm from the library was used for the data scikit-learn [11] K-means. As a result of applying the k-means method on the training set, we obtained 8 clusters. On Figure 6.1 there are visualisation of cluster analysis results. Clusters are visually separable from each other.

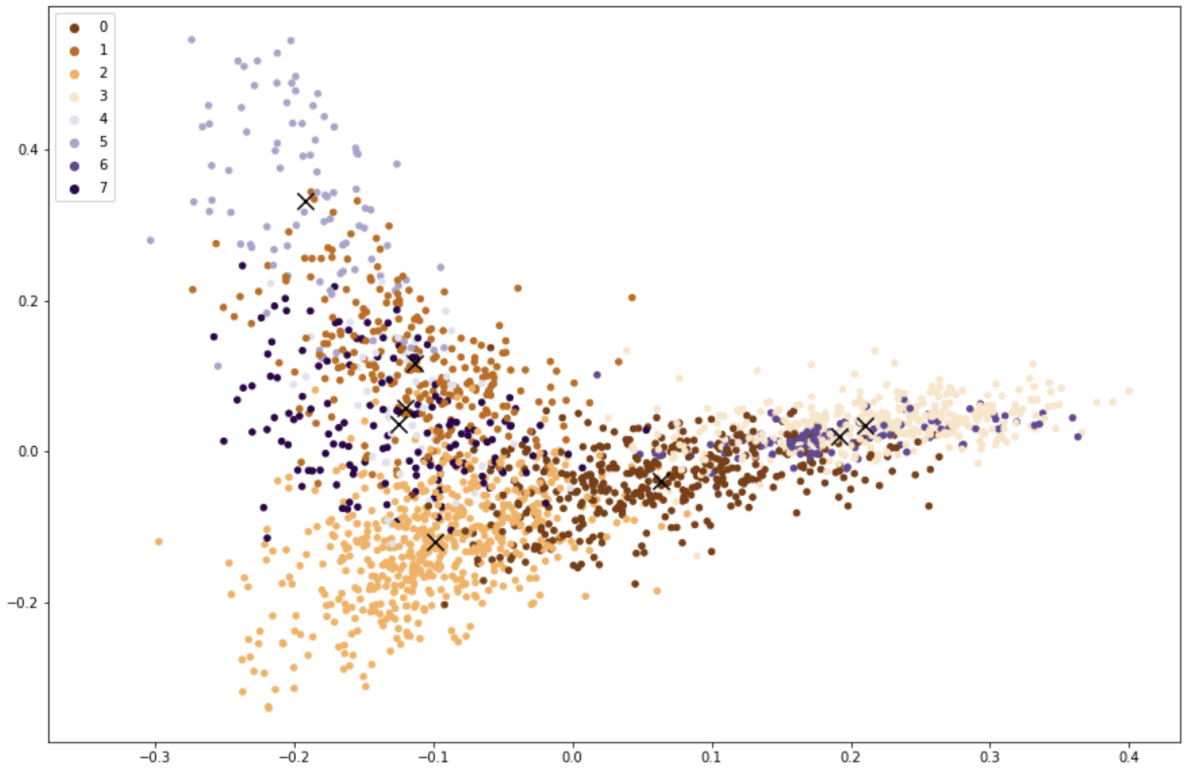



Figure 6.1 — Kmeans. 8 clusters

Further, to try to improve the model, the pre-trained Bert model for the Russian language was used. However, when using the K-means method together with the pre-trained Bert model, the quality of clustering turned out to be significantly lower than just using the K-means method. On Figure 6.2 there are results of Kmeans method with Bert. The clustering result of this method are significantly lower in comparison with Kmeans method, since it is difficult to understand results, all the data is in one place. Therefore, further analysis of the results for a simple K-means method will be performed.

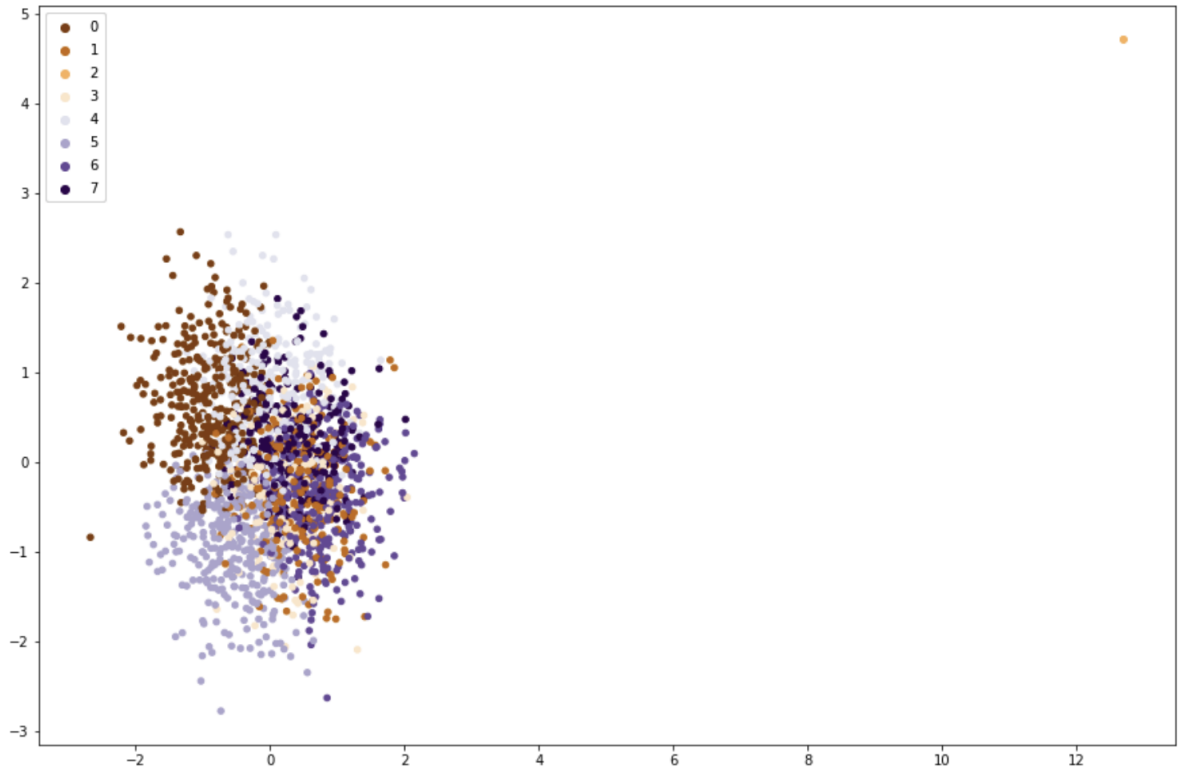



Figure 6.2 — Kmeans + Bert. 8 clusters

## 6.2 Clustering Results

The list of keywords by cluster is provided below:

— Cluster 0: получение, композиции, кислоты, композиция, мас, основе, способу,материал, изготовление, газ, смеси, слой;

— Cluster 1: данные, изображения, информация, блок, кодирование,декодирование,множество,основа, сигнал, первый, устройство, передача;

— Cluster 2: элемент, часть, устройство, поверхность, аэрозоль, узел, направление, корпус, элемент, отверстие;

— Cluster 3: лечение, антитело, клетки, рак, человек, заболевание, применение, последовательность,композиции,против;

— Cluster 4: транспортный, средство, система, панель, управление, камеры, достигается, множество;

— Cluster 5: связь, беспроводной, передача, линия связи, доступ, управления, сети, ресурсы, устройство, канал, данные, информация;

— Cluster 6: алкил, формула, группа, соединение, выбран, состоящий, каждый, галоген;

— Cluster 7: управления, двигателя, система, сгорания, клапана, системы, датчика, давления, мощности, датчик, устройства, выполнен, внутреннего, определения, сигнала;

Based on this, the following distribution of clusters by regions can be made:

— Cluster 0 - Manufacturing of materials, composite materials, mixtures.
— Cluster 1 - Data and image processing.
— Cluster 2 - Creation of aerosols for surfaces and devices.
— Cluster 3 - Medicine, cancer and disease treatment
— Cluster 4 - Transport, transport management, cameras and systems for transport.
— Cluster 5 - Communication, access and data transmission devices.
— Cluster 6 - Chemistry, compounds, halogen group and alkyl group.
— Cluster 7 - Creation of motors and sensors for them.

Also on Figure 6.1, you can see that similar areas are located side by side. For example, Medicine (cluster 3) and Chemistry (cluster 6)

The next step was to evaluate the model's performance on new data. For this, a sample was previously formed and cleaned on new 2000 texts. Also,this validation sample were manually labeled for result estimation.

The distribution by class in two samples turned out as follows: The proportions of the ratio of data to the classes provided on table 6.1.

| Cluster | True Data | Cluster Data |
|---|---|---|
| Cluster 0 | 194 | 213 |
| Cluster 1 | 208 | 199 |
| Cluster 2 | 102 | 234 |
| Cluster 3 | 229 | 231 |
| Cluster 4 | 44 | 44 |
| Cluster 5 | 228 | 190 |
| Cluster 6 | 330 | 269 |
| Cluster 7 | 348 | 273 |

Table 6.1 — Data distribution by clusters

It is noticeable that apart from the significant difference for Cluster 2 and Cluster 6, the rest of the clusters have a relatively similar distribution. As a result, only 30 texts turned out to be non-clustered data in the new sample.

Let's calculate the F1 metric for new data according to the approach described in the article "Evaluating Clustering Results" [8].

F1-score = 74.9%

### 6.3 Folder with code and data

**link :**

https://github.com/yaxenia/Text-Clastering

— google_patents.csv - result from google patents service

— patents.csv - downloaded patent texts

— clusters.csv - is the data on which the model was trained with the clustering result

— validation.csv - new data sample with clustering results

— frequensy.csv - result of frequency analysis converter.py - code parse data from google_patents.csv and the web links of files and extract text from the pdf file.

— cluster.ipynb - code with preprocessing of data and clustering

# Conclusion

Based on the results of the work, the following tasks were solved:

— Study of the relevant theory and search for similar solutions to the text classification problem

— Formation of a database consisting of patent texts.

— Data preprocessing and analysis

— Clustering for patents

— Analysis and comparison of the results obtained

In this work, the data of patents were taken, the texts were preprocessed and cleared of stop words and punctuation symbols. Also, as a result of this work, a model was trained, which clustered a set of patent texts for 8 clusters. This model was tested on two samples and showed a fairly good performance result. An analysis of the results was carried out, with the help of which it was possible to determine the keywords for each cluster and the scope of the clusters:

— Cluster 0 - Manufacturing of materials, composite materials, mixtures.

— Cluster 1 - Data and image processing.

— Cluster 2 - Creation of aerosols for surfaces and devices.

— Cluster 3 - Medicine, cancer and disease treatment

— Cluster 4 - Transport, transport management, cameras and systems for transport.

— Cluster 5 - Communication, access and data transmission devices.

— Cluster 6 - Chemistry, compounds, halogen group and alkyl group.

— Cluster 7 - Creation of motors and sensors for them.

.

To assess the quality of clustering on a new validation set with manually labeled clusters, the F1-score metric was used, the result was as follows: F1-score = 74.9%

In conclusion, we can say that this work has achieved its goal.

# Bibliography

1. Aleksandr Kukushkin. *Natasha parser. Project of Russian language processing.* https://github.com/natasha

2. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze *Introduction to Information Retrieval.* Cambridge University Press, 2008. p: 253-285. https://nlp.stanford.edu/IR-book/pdf/13bayes.pdf

3. Radim Řehůřek's *Deep learning with word2vec and gensim* RARE TECHNOLOGIES, 2013 https://rare-technologies.com/deep-learning-with-word2vec-and-gensim/

4. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* Cornell University, 2018

5. Фомин В.В.,Флегонтов А.В.,Осочкин А.А. *Метод частотно-морфологической классификации текстов* Российсикй государственный педагогический университет имени А.И. Герцена, 2017

6. *Natural Language Toolkit* https://www.nltk.org

7. *ML | Determine the optimal value of K in K-Means Clustering* Geek for geeks, 2019

8. Vijini Mallawaarachchi *Evaluating Clustering Results* Towards Data Science, Jun 9, 2020. https://towardsdatascience.com/evaluating-clustering-results-f13552ee7603

9. https://patents.google.com

10. https://stanfordnlp.github.io/stanfordnlp/

11. https://scikit-learn.org/stable/

12. https://www.wipo.int/classifications/ipc/ipcpub