Report

Introduction

The project aims to work with text data, specifically, preprocessing and clustering it into 5 groups. The report consists of four main parts: first stage involves the creation of basic model to accomplish the task, in the next part more advanced solutions are tested. In the third part the results are compared and content analysis for best model is made. Finally, instructions for execution of the project code are provided.

Appendix with Figures is shown at the end of the report.

1. Baseline solution

1.1 Data description

Looking at provided data is needed before starting to build any models. The data consists of 1332 texts, each text is described with four main attributes: «id», «class», «title», «abstract». «Class» takes values from 1 to 5, providing information about the original class of each text, it is used to check the results of clustering model work. «Title» represents the heading of each text and «abstract» contains the text itself.

As shown in Figure 1.1, the first class contains 353 texts, it is the largest value among all the others, least represented category is the second one – it has 202 texts.

1.2 Preprocessing

In the basic model, elementary preprocessing has been done. The title and the abstract have been combined for each text. After that tokenization, which represents the text as list of strings, takes place. Also, all words are lowercased. These stages are basic ones and are applied for all other models too.

Set of stop words includes english words, also from the very beginning some punctuation signs have been added to it. Stop words are deleted from texts, after that stemming with PorterStemmer() [1] from nltk toolkit is done.

1.3 Representation

Data is represented in TF-IDF form (it is used in all other models). As sklearn library is used [2], smoothed IDF weights are obtained. Also, every output row has unit norm, calculated with l2 normalization. The mathematical expression for used TF-IDF representation is shown in Equation 1 [3].

$$tfidf(w_j, d_i) = fr(w_j|d_i)\left(log\left(\frac{1+n}{1+n_j}\right) + 1\right) \quad (1)$$

, where

w_j - term

d_i – document

n_j – number of document, where w_j occurs

The transformed matrix has 1332 rows and 9646 new feature columns.

1.4 Distance function

In that case, data is normalized, Euclidean distance can be used due to Equation 2.

$$L_2^2(x1, x2) = 2\big(1 - cos(x1, x2)\big) \quad (2)$$

1.5 Clustering method

In the basic approach, ordinary K-Means is used with 5 clusters and Euclidean distance.

1.6 Quality measure

As in the task, the results have to be compared with NMI, it is chosen as the quality measure to obtain optimal parameters. It is used for all models. The Equation 3 provides the formula, used to calculate the score.

$$NMI = \frac{I(C,D)}{\sqrt{H(C)H(D)}} \quad (3)$$

, where

$I(C,D) = \sum_{C_i \in C} \sum_{D_j \in D} P\big(C_i, D_j\big) log \frac{P\big(C_i, D_j\big)}{P(C_i)P\big(D_j\big)}$,

$$H(C) = -\sum_{C_i \in C} P(C_i) log P(C_i)$$

The final NMI score for basic clustering is 0.72. Better results can be obtained by using more advanced approaches.

2. Advanced solutions

2.1 First model

2.1.1 Preprocessing

In more advanced approaches, special attention is paid for preprocessing stage. If the data quality is low, the algorithm has small chances to perform well, so that step is one of the most important in the task and affects the final results significantly.

At first, title and abstract features are combined, and tokenization is performed. Words are made lower, english stop words and possible punctuation symbols are removed.

Instead of stemming, lemmatization is used, which in general seems to be better, more accurate approach to work with data. Also, numbers, possible url-links are removed. And only those words are left, whose length is larger than 3. After that 150 most frequent words have been checked to find some unnecessary ones. Words, like, «many», «using», «used», «well» have been deleted, as at that stage it has been decided that they do not bring any value.

2.1.2 Representation

The texts are again presented in TF-IDF form (the same as in 1.3), but here the maximum number of features is set to 800 (different values near 1000 have been checked, 800 is the best one in that case). Top features are chosen by ordering them with term frequency across the corpus. Also, n-grams range is set from 1 to 3, that means that not only single words, but also 2 or 3 of them can be combined and used.

2.1.3 Distance function

In that case, data is normalized, Euclidean distance can be used due to Equation 4.

$$L_2^2(x1, x2) = 2\big(1 - cos(x1, x2)\big) \quad (4)$$

2.1.4 Clustering method

In that approach, at first ordinary K-Means is used with 5 clusters and Euclidean distance.

Secondly, Spectral clustering is used with 5 clusters.

2.1.5 Quality measure

NMI for K-Means is equal to 0.73.

NMI for Spectral clustering is 0.69.

As we can see, the NMI for K-Means has not increased much after advanced preprocessing and lemmatization. The score for spectral clustering is a bit lower than for K-Means. After obtaining that results, it was decided to try the same preprocessing, but with stemming and analyze the obtained results.

2.2 Second model

2.2.1 Preprocessing

Here exactly the same preprocessing is used as in 2.1.1., however, instead of lemmatization – stemming is used.

2.2.2 – 2.2.3

Same as in 2.1.2.-2.1.3

2.2.4 Clustering method

K-Means and spectral clustering have been tried as in the previous model. Data is divided into 5 clusters and final results are obtained.

2.2.5 Quality measure

NMI for K-Means is equal to 0.76. In increased a bit.

NMI for spectral clustering is 0.73. It increased too.

That tells us that maybe stemming for that type of data is better, than lemmatization. We will stick to this approach further on.

As the results of K-Means are rather good here, PCA with 2 components is made to plot the data. In Figure 2.1 it is clearly shown that some clusters are well-separated, others are a bit mixed, but overall, the quality is satisfying.

2.3 Third model

2.3.1 Preprocessing

In that case same preprocessing, as in 2.2.1 and 2.1.1 is used: with advanced data cleaning and stemming.

2.3.2 Representation

The texts are again presented in TF-IDF form, but now there is no limit for number of features. Also, n-grams range is set from 1 to 3, that means that not only single words, but also 2 or 3 of them can be combined and used.

2.3.3 Feature extraction

A conventional approach of LSA (realized with TruncatedSVD) is used to transform document-term matrices to a «semantic» space of lower dimensionality. It can possibly help with synonymy and a bit with polysemy.

2.3.4 Distance function

Again, Euclidean distance is used for future K-Means and normalized data.

2.3.5 Clustering method

In that approach, ordinary K-Means is used with 5 clusters and Euclidean distance.

It is tried for several LSA decompositions with different number of components.

2.3.6 Quality measure

NMIs for K-Means clustering with different LSA decompositions are shown in Figure 2.2. As it can be seen, NMIs are higher, than in the previous cases and there are some, which are larger than 0.81. The largest NMI value of 0.8221 is obtained for clustering of SVD matrix with 50 most valuable components. It is rather good score, larger, than 0.81 threshold. It was decided to leave it as the best variant and analyze it.

3. Best model choice

3.1 Models comparison

Built models are compared in Table 1.

Model	**Parameters**	**NMI**
Model 1 - baseline	Basic preprocessing Stemming K-Means	0.72
Model 2.1.1	Advanced preprocessing Lemmatization Tf-Idf with max_features and n-grams K-Means	0.73
Model 2.1.2	Advanced preprocessing Lemmatization Tf-Idf with max_features and n-grams Spectral	0.69

Model 2.2.1	Advanced preprocessing Stemming Tf-Idf with max_features and n-grams K-Means	0.76
Model 2.2.2	Advanced preprocessing Stemming Tf-Idf with max_features and n-grams Spectral	0.73
Model 2.3	Advanced preprocessing Stemming Tf-Idf with n-grams LSA 50 components K-Means	0.8221

Table 1

3.2 Best model content analysis

The best model is done by K-Means clustering of LSA decomposition with 50 main components. That is why that best model's results are analyzed.

There are 5 clusters and for each of them top frequent five words have been found, as well as the main topics, like in Table 2. As top words have been stemmed, they have been looked at the original texts and here lexically full top words are shown.

Cluster number	Top word	Frequency	Topic
1	database data relational system queries model information approach propose process	681 583 361 308 270 192 181 155 137 133	That topic is connected with database development, database schema designs and database management systems theory overall. Knowledge organization in database is also spoken about.
2	secure propose scheme encryption data algorithm cryptography computer system protocol	676 391 345 318 317 251 251 231 229 228	The topic is connected with information privacy, security (cybersecurity), data encryption.
3	robot control system model perform propose result	771 325 286 169 167 143 143	For that topic, the most common words are connected with intelligent robotics and automation systems, system controlling development and design.

	task develop design	138 128 125	
4	compiling program computer code language optimizer system generate implement paper	592 384 300 260 243 213 212 181 178 174	Topic is mostly connected with theoretical computer science: compilers, programming languages, optimization algorithms, quantum computing. Computational, program architecture is discussed.
5	image method detect model propose compute system learn perform network	618 563 485 470 456 388 386 331 328 323	That topic is connected with image analysis, pattern detection modelling. Documents are mainly addressing the sphere of computer vision and pattern recognition.

Table 2

4. Execution instructions

The Jupyter notebook with the code to run the program is attached to the submission.

At first, there is section with imported libraries, needed to run the program, they have to be installed on the device. Also, under the «Download resources section» commands for downloading nltk and possible other toolkits on the computer are listed.

The main part of the notebook is divided into several parts, corresponding to the structure of that report. At first – data is explored. After that comes the part with baseline solution, then – part with possible advanced solutions. Finally, the discussion about the best obtained results is provided.

References

1) NLTK documentation and manual. [https://www.nltk.org/howto/stem.html].
2) Sklearn documentation. [https://scikitlearn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html]
3) Methods of Data Mining course, Aalto University, 2021. Lecture 11.

Figures

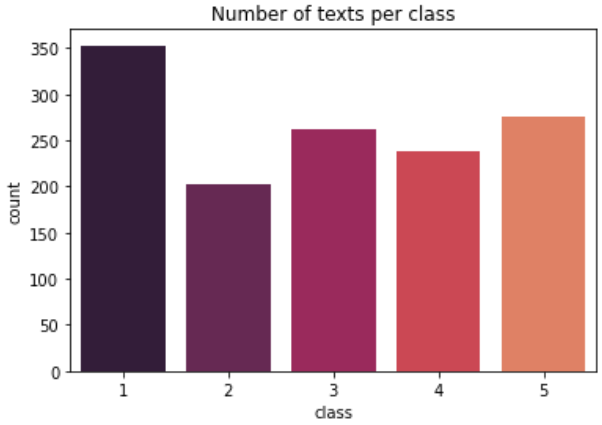

Figure 1.1 Number of texts per each of 5 classes

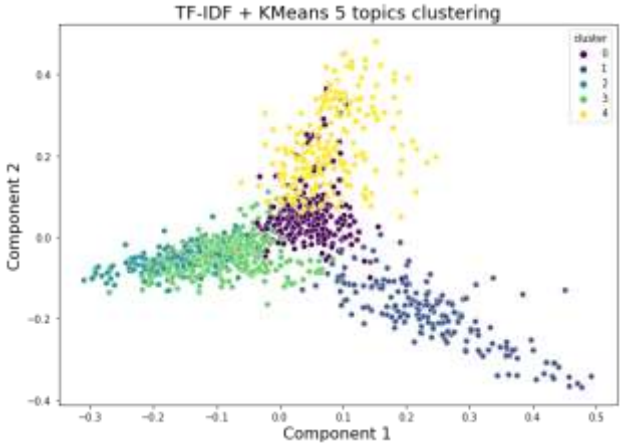

Figure 2.1 Number of texts per each of 5 classes

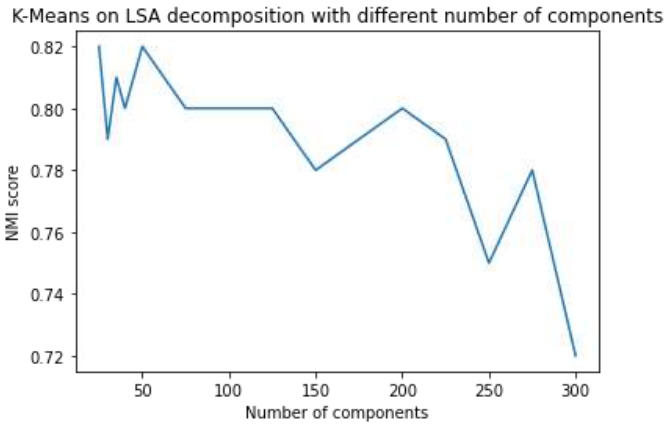

Figure 2.2 NMI scores for K-Means clustering with LSA decomposition