Data Science Project Competition Automated Customer Email Classification

Miha Likar and Neža Marija Slosar

Abstract

Often emails must first be assigned to a category and sometimes even subcategories before we can determine who should respond and how. Our task is to make such a classification in (pre)determined categories. Firstly we tokenized emails and made some cleaning of texts. Second we adjusted those features for Slovene language. Third we visualize data and analyzed the trend. Fourth we tried some models and found out that logistic regression works best. Currently we are working on clustering, based on Slovene, finding most optimal categories and best deep learning model for autonomous classification.

Keywords

Email classification, Natural language processing, Slovene, Supervised classification, Deep learning models

Advisors: doc. dr. Slavko Žitnik, Andrej Mišič

Introduction

Email classification is one of the first step in good customer support, because decrease responding time and also open space for employees to spend time instead of manually classifying email on other more relevant tasks. [1][2]

Email classification is based on artificial intelligence, combining linguistics and machine learning. Based on huge datasets, which are cumulating on the internet, new sufficient approaches in natural language processing (NLP) are developed. Still is important to mention the aspect of language itself. We are working on Slovene emails, so, we should take all the differences from English into account. On that point we can help with CLARIN Slovenia - Common Language Resources and Technology Infrastructure, Slovenia, which is pioneer on Computer and Corpus Linguistics, and Digital Humanities. Researchers, working on that project had also developed classla, pipline from Python NLP library Stanza.

Further more email classification is not new thing, it began in clustering *ham* and *spam* emails. [3] Nowadays lot of different learning models are widely used: BERT (Bidirectional Encoder Representations from Transformers); DNN (deep neural network) with BiLSTM (bidirectional Long Short Term Memory) layer; classifier k-NN (k-nearest neihbours) and NB (Naive Bayes). [3]

Based on all these knowledge, models and NLP Course from our supervisor doc. dr. Slavko Žitnik, our aim is to develop autonomous email classifier and make faster and more efficient customer support.

Methods

We got data (emails) separated by months (from January 2021 till June 2021), first we merge them together in one data frame. Then we clean not labeled categories and we got all together 35.698 emails, separated in 15 categories. Unlabeled emails can be used as additional testing data or will be used for some clustering process.

Firstly we started with a general preprocessing procedure, which included lowercasing, punctuation removal, digit removal, email adresses removal and web links removal. These processes were made by applying regex functions. Secondly we have also applied stopwords removal based on Slovene library. After the first analysis we have found out that some uninformative words remained in the data, so we made list of additional uninformative words based on this primary analysis. In the second round of preprocessing we have also used classla, mainly to achieve lemmatisation. We used KMeans clustering from sklearn library and are still in the process of visualisation with the help of seaborn. The original number of designated categories in was 15, so we have decided to test a range of clustering the data provided from 20 to 8 clusters. The process of visualisation is quite time consuming since we have to perform the PCA (principal component analysis) on

the multidimensional data, so we are still in the process.

Since the vocabulary in the emails is not that rich, a lot of informative words were recognized as different words and this had a negative impact on the models. Lemmas expectedly turned out to be much more efficient in both the model training and testing, as in the case of clustering as well. Thirdly, we count out, how many emails are in each category. We found out, that “Marketinške akcije” and “GDPR” have under 55 emails, “Banke” and “CORIS” under 20 and “Zavarovanja - zunanji partnerji”, “Help desk - interna javnost” under 10 emails. Based on that we expected that this amount of emails will not be enough informative word extraction and consequentially for model training and testing. Fourthly we visualize data and put them in first examples of learning models.

After that we did clustering with classla and based on that data ran few models:

- dummy classifier,
- naive Bayes,
- linear support vector classifier and
- logistic regression.

Figures

One of examples of visualization is plot with frequencies of words – Figure 1. It is very clear sight on what could be key words for specific category.

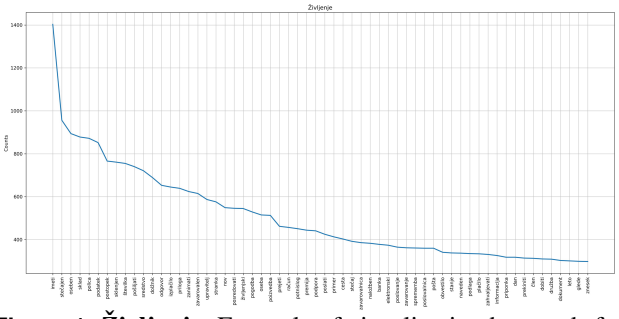

Figure 1. Življenje. Example of visualization by words for category Življenje.

Code examples

We provide here part of preprocess code, which shows cleaning and tokenizing data. Furthermore we are showing part of logistic regression code, because is regarding to classification accuracy best working model.

Listing 1. Preprocessing of data.

```
def preprocess(df):

    #SPREMENI NaN V CLASS "BREZ"
    df = df.fillna("Nedefinirano_polje")
    #VR I IZ DF VSE, KI IMAJO CLASS "BREZ"
    df = df.loc[~df["class"].str.contains("
        Nedefinirano_polje")]

    #VR I VEN MAILE BREZ VSEBINE
    df = df.loc[~df["content"].str.contains("
        Nedefinirano_polje")]

    #Vr i ven "unnamed
    df = df.drop(columns=["Unnamed:␣0"])

    #REMOVING WEBSITES FROM STRINGS
    #df['tokenized'] = df['tokenized'].apply(
        lambda x: re.sub(r'^https?:\/\/.*[\r\n]*',
         '', x))
    df['tokenized'] = df['content'].apply(lambda x
        : re.sub(r"(https?:\/\/)(\s)*(www\.)?(\s)
        *((\w|\s)+\.)*([\w\-\s]+\/)*([\w\-]+)((\?)
        ?[\w\s]*=\s*[\w\%&]*)*", '', x))

    #REMOVING EMAIL ADRESSES FROM STRINGS USING
        REGEX
    df['tokenized'] = df['tokenized'].apply(lambda
         x: re.sub('\S*@\S*\s?', '' , x))

    #LOWER-CASING NO_STOPWORDS COLUMN
    df["tokenized"] = df["tokenized"].str.lower()

    #STRIPPING NO_STOPWORD COLUMN
    df["tokenized"] = df["tokenized"].apply(lambda
         x: x.strip())

    #REMOVING PUNCTUATION
    df['tokenized'] = df['tokenized'].apply(lambda
         x: re.sub('[%s]' % re.escape(string.
        punctuation), '␣' , x))

    #REMOVING DIGITS
    df['tokenized'] = df['tokenized'].apply(lambda
         x: re.sub('[\d]', "" , x))

    #REMOVING SINGLE CHARACTERS
    df['tokenized'] = df['tokenized'].apply(lambda
         x: re.sub(r"\b[a-zA-Z]\b", "" , x))

    #REMOVING WORDS LONGER THAN 16 CHARACTERS FROM
    df['tokenized'] = df['tokenized'].apply(lambda
         x: re.sub(r'\b\w{16,100}\b', "" , x))

    #REMOVING WORDS 1-2 CHARACTERS FROM
    df['tokenized'] = df['tokenized'].apply(lambda
         x: re.sub(r'\b\w{1,2}\b', "" , x))

    #DETERMINE SLOVENE STOPWORDS
    slo_stopwords = stopwords.words("slovene")
    all_stopwords = slo_stopwords + add_stopwords

    #REMOVE STOPWORDS FROM SLOVENE BASE
    df['tokenized'] = df['tokenized'].apply(lambda
         x: '␣'.join([word for word in x.split()
        if word not in (all_stopwords)]))

    df = df.replace(r'^\s*$', np.NaN, regex=True)

    #SPREMENI NaN V - post-classla - "Nedefinirano
         polje"
    df = df.fillna("Nedefinirano_polje")
    #VR I VEN MAILE BREZ VSEBINE - post-classla
    df = df.loc[~df["tokenized"].str.contains("
        Nedefinirano_polje")]

    df.reset_index(drop=True, inplace=True)

    #KLASLA
    df["tokenized"] = df["tokenized"].apply(lambda
```

```
        x: klasla(x))


    return df
```

Listing 2. Clustering data.

```
import matplotlib.pyplot as plt

true_k = 10
model = KMeans(n_clusters=true_k)

model.fit(vectors)

order_centroids = model.cluster_centers_.argsort()
    [:, ::-1]
terms = vectorizer.get_feature_names()

clusters = model.labels_.tolist()
print("Clusters: {}".format(clusters))


with open ("clustering_post_klasla.txt", "w",
    encoding = "utf-8") as f:
    for i in range (true_k):
        f.write(f"Cluster {i}")
        f.write("\n")
        for ind in order_centroids[i, :20]:
            f.write(" %s" % terms[ind],)
            f.write("\n")
        f.write("\n")
        f.write("\n")
```

Listing 3. Logisitc regression model.

```
%%time

#LOGISTIC REGRESSION

from sklearn.linear_model import
    LogisticRegression

logreg = Pipeline([('vect', CountVectorizer()),
                ('tfidf', TfidfTransformer()),
                ('clf', LogisticRegression(n_jobs
                    =-1, C=10,solver='lbfgs',
                    max_iter=100,
                                        )),
               ])
logreg.fit(X_train, y_train)


y_pred = logreg.predict(X_test)

print('accuracy %s' % accuracy_score(y_pred,
    y_test))
print(classification_report(y_test, y_pred,
    target_names=my_tags))
```

Results

From preprocessing we find out, which words have potential to be key words of specific category, we visualize those data to get insight, if our preprocessing make sense and provide useful information to further work with them on models. After using classla for additional processing we got far better results of word frequencies in lemmatisation. We are expecting much better results in clustering process using the lemmas.

Afterwards we found out from training models, that categories: “Marketinške akcije”, “GDPR”, “Banke”, “CORIS”, “Zavarovanja - zunanji partnerji” and “Help desk - interna javnost” does not have enough data to provide efficient classification accuracy. We will agree later in process on best solution for those categories.

Test scores from dummy classifier are:

1. most frequent: 0.43454193191253804,
2. stratified: 0.2891410646738629,
3. uniform: 0.06771842420887536,
4. constant: 0.43454193191253804.

Tables

We made example of frequency table, which is first inside into category understanding and visualizing of key words.

Table 1. Frequency table for Življenje category.

Word	Frequency
imeti	1403
stečajen	956
oseben	894
sklad	878
polica	872
podatek	852
postopek	766
sklenjen	761
številka	755
pošiljati	740
sredstvo	721
dolžnik	689
odgovor	653
izplačilo	645
priloga	639
zanimati	624
zavarovalen	615
upravitelj	587
stranka	576
nov	549

Second we are reporting accuracies from models. Naive Bayes provide:

Table 2. Naive Bayes results.

	precision	recall	f1-score
accuracy			0.72
macro avg	0.35	0.18	0.19
weighted avg	0.74	0.72	0.65

Linear support vector classifier report slightly better results compare to naive Bayes.

Logistic regression model perform best from all models, with accuracy of 0.82. So, we conclude to work on with this model.

Table 3. Linear support vector classifier results.

	precision	recall	f1-score
accuracy			0.77
macro avg	0.37	0.26	0.27
weighted avg	0.76	0.77	0.72

Table 4. Logistic regression results.

	precision	recall	f1-score
accuracy			0.84
macro avg	0.63	0.51	0.56
weighted avg	0.83	0.84	0.83

Discussion

After this first dive into NLP we decide at the beginning to understand and visualize data. At that point we recognized the first obstacle, which is not enough data for some categories and heavier challenge in NLP of Slovene language.

We have pretty much finished with preprocessing, regarding tokenizing, lemmatizing, cleaning and clustering data. Second we did first meaningful visualizations. Additional work shall be done regarding the clustering since the process of lemmatisation was finished recently. Third we tried few basic learning models, to see if preprocess works well and what should we take into consideration for other models.

Based on learning outcomes we are going to work on two directions for next few months. First, we will prepare reasonable categories and do on data as much as we can with classla, regarding clustering and other features. Second, we will try to use apart logistic regression some other, according to literature more powerful models – we are going to start with BERT.

Our main goal for next few months is to develop a code, which will do very good clustering from the point email step into the system, and second connect this to classification model, which will classify email (based on a bit different categories from now) with at least 0.9 classification accuracy.

Acknowledgments

We would like to thank our supervisors – Slavko Žitnik and Andrej Mišič, who are giving us support and help us with all NLP understanding. We would also like to thank Jure Demšar and his team for marvelous work on organisation and last but not least, we are thankful for support from Zavarovalnica Triglav, where we can always get useful information, regarding data and concepts.

References

[1] Michael Wenceslaus Putong and Suharjito Suharjito. Classification model of contact center customers emails using machine learning. *Advances in Science, Technology and Engineering Systems Journal*, 5(1):174–182, 2020.

[2] Anton Borg, Martin Boldt, Oliver Rosander, and Jim Ahlstrand. *E-mail classification with machine learning and word embeddings for improved customer support*. Blekinge Institute of Technology, Master's thesis, 2017.

[3] Isra AbdulNabi and Qussai Yaseen. Spam email detection using deep learning techniques. *Procedia Computer Science*, 184:853–858, 1 2021.