# Improvement of a Ticketing System

## Abstract

In the context of the advanced practical course "Machine Learning for Information Systems Students" our task was to improve the Ticketing System of UCC company. During the project, we introduced machine learning algorithms to improve the system to meet the needs of both the users and the company as much as possible. Our main task is to use machine learning algorithms to cluster the problem descriptions initially entered by users. First of all, we separated the text information from the dataset, and then preprocessed the data for different languages (mainly English and German) after language classification. We then built the models. After building the model we carried out post-analysis and model evaluation, and finally identified possible future directions.

# 1 Background ＆ Problem

## 1.1 Demand Analysis

### 1.1.1 Background & Purpose

Ticketing system is a very important channel for companies to get feedback and provide support to users. Nowadays most companies on the market have their own support systems. Our project is for UCC's support system.

The UCC is an education-as-a-service company at the TU Munich that provides SAP services (hosting, support, training) to educational institutions (e.g., other universities). It internally uses an SAP ticket system based on a standard configuration to handle incoming customer requests. Requests can be submitted by creating a ticket in the system.

The overall goal of this project is to identify ways to improve this support system, with the help of Machine Learning, to improving classification accuracy, reduce the time and effort needed from the staff and improve customer experience.

### 1.1.2 Demand Analysis Process

**User Report Process**

Before
Ticket System
After

Figure 1.1: 3 parts of problem-report

Before report: users find the report-portal and prepare to submit problems or terminate report directly by the FAQs.

During report: users submit problems by using the Ticketing System, including the information about the problems, such as time of failure, the content of problems, contact details, etc.

After report: the company solves the problems by communicating with users and also can use this way as a flow inlet to invite users to take satisfaction surveys.

The specific process is illustrated below:

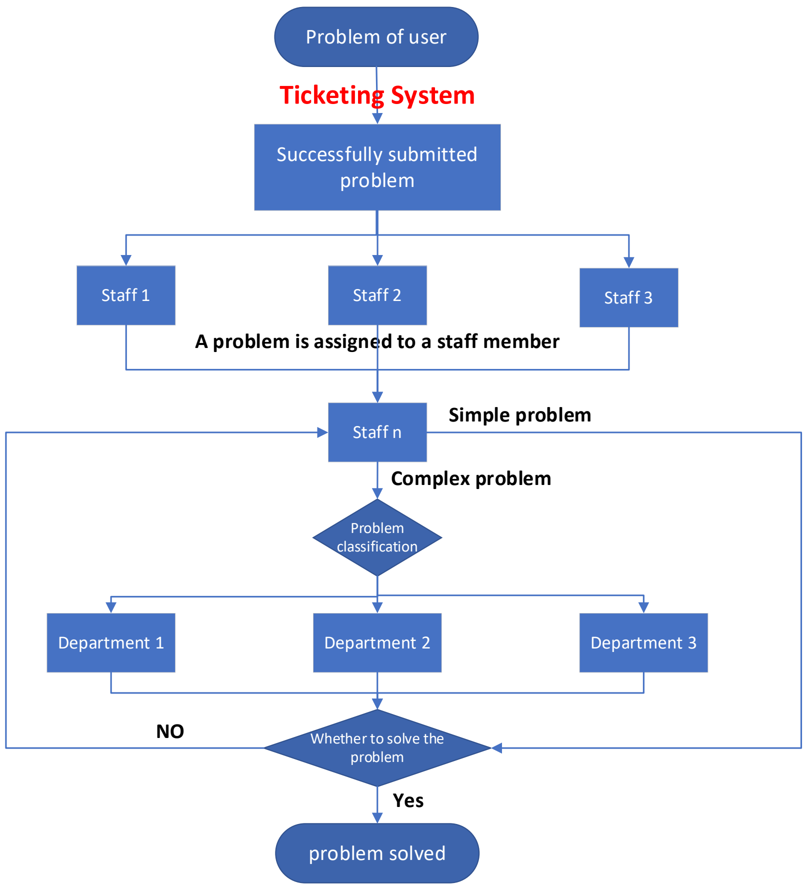


Figure 1.2: Support system flow diagram

### Existing Problems with the System

1) The existing Ticketing System cannot efficiently and automatically classify/cluster users' problems within a short period of time and assign them to professionals to solve them. This consumes a lot of labor and time costs of the company.

2) Users often do not look at FAQs before submitting a support request. This could be due to multiple reasons: poor quality, irrelevant or outdated answers to questions, FAQs being hidden in a subpage of the website, etc. The topic of FAQs warrants extra analysis, and highly relevant and visible FAQs could be one pillar in the overall customer support strategy.

**Demand from Both Sides**

For users:

1) A smooth and unobstructed problem-report process.

2) Problems can be solved quickly and correctly.

For UCC company:

1) Staff can receive clear descriptions of problems from users in a timely manner, quickly solve simple problems and assign complex problems correctly and quickly to the appropriate department.

2) The problem handling process is efficient and simple.

3) Saving costs.

4) Improving user satisfaction.

### 1.1.3 Literature Review and Analysis for Competitive Products

Due to the heavy workload of distributing and processing user tickets, the cost of supporting the company is increased greatly and the time for problem resolution is also extended. The advancing potential of Machine Learning algorithms has led to the goal of automating those support ticket systems, enabling the ticket system to classify tickets without manual intervention. Therefore, Machine Learning algorithms are tried to be introduced into the ticket system to complete the task of automatic classification. At present, the main algorithm is supervised learning. Through reading literatures in related fields, we found that ML algorithms such as Random Forrest and Support Vector Machines currently perform best in ticket-classification. In recent years, there are also some literatures which indicate that Deep Learning algorithms will play an important role in this field shortly.

Taking the application of Machine Learning in Amap's user-feedback support system as an example: Users usually use the ticket support system on the mobile phones and PCs to report their problems. When reporting, users can select some options and text descriptions to report problems. The development team divides the tickets entered by users into two classes, respectively a class with invalid description and the other with valid description. Tickets with invalid descriptions are directly transferred to manual processing. For valid-described tickets, the development team uses Machine Learning algorithms to classify the tickets. After preprocessing, the word vectors of user-report information are used as input, and the output, namely class labels, needs to be manually labeled. Therefore, this multi-class supervised learning model is based on obtaining a dataset which has sufficient number of classified tickets with correct labels. In addition, LSTM is selected to build the model. The LSTM is a kind of Deep Learning algorithm that can well solve the problems of gradient disappearance or gradient explosion during the training process. The development team also applies different optimization methods to improve the accuracy of the neural network. For the problem of insufficient accuracy of a single model, the Ensemble method is introduced to solve the problem.

Due to the specificity and individual characteristics of each company's ticket system, each system has different requirements when Machine Learning algorithms are introduced. The user-data currently collected by UCC's ticket support system lacks accurate classification labels. For supervised learning, the accuracy of the labels directly determines the quality of the model. Based on this consideration, at this stage, we decide to establish an unsupervised learning model, such as the K-means clustering algorithm.

### 1.1.4 Conclusion

We can introduce Machine Learning algorithms to improve the system and meet the needs of users and companies. The optimization scheme we propose is to isolate the problem description input by the users for the first time, extract the effective information from it, and use the clustering algorithm to provide analysis and basis for subsequent problem classification, and mine ticket text information. For example, the number of clusters is 2, one cluster is the problems with specific description which can be directly assigned to different departments, another is the rest problems with ambiguous description which should be assigned by supporters.

## 1.2 Overview and Preliminary Analysis of Data

There are 12,184 user samples in total. There is a ticket summary file that includes the attribute information of all tickets, a status file that summarizes the status changes of all tickets, and all tickets with the specific content sent by the users.

### 1.2.1 Ticket Summary File

The ticket summary file includes the following attribute information about all tickets:

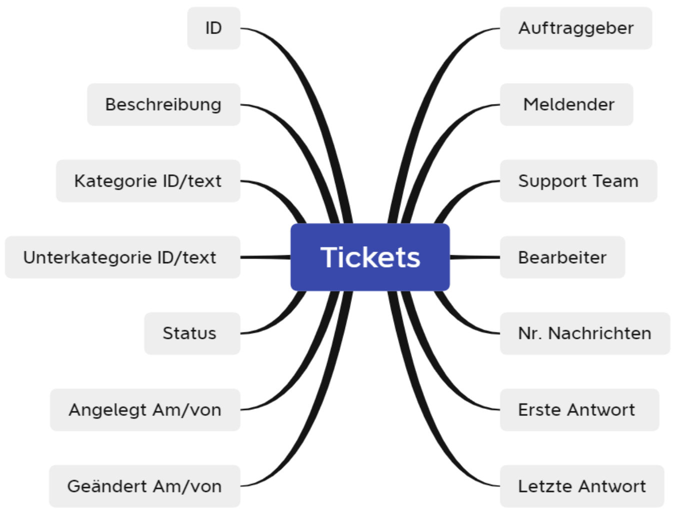


Figure 1.3: Attributes of the tickets

Since a lot of information in the form is input by users, due to the user's misunderstanding or inaccurate description of the problems, typos, etc., there is a lot of wrong information in the form. In addition, a large amount of data is missing. If we use the data directly, the accuracy of the model cannot meet the requirements. In addition, by filtering and analyzing the attribute "Angelegt von" of the tickets, it can be found that "SOLMAN_BTC" users have generated 5529 Alert tickets. In addition, there are also some Alert tickets sent by other users. These tickets are very close and are resolved by several dedicated support teams (44, 46, 47). Those tickets are automatically generated to inform the technical crew about minor issues regarding the ticket system itself.

### 1.2.2 Status File

The status file includes the status information of all tickets:

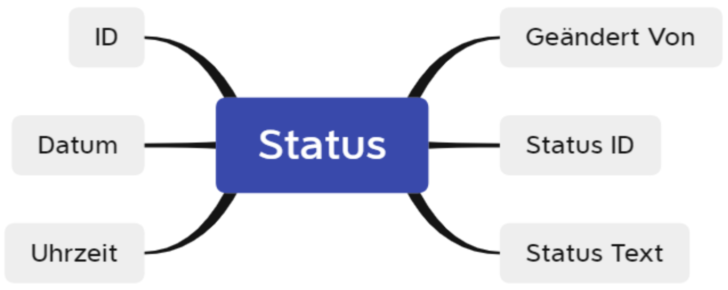


Figure 1.4: Attributes of the status

The model trained by our project does not need to use status-related information, so we won’t go into details.

### 1.2.3 Tickets of Users (main)

The content of each ticket is as follows:

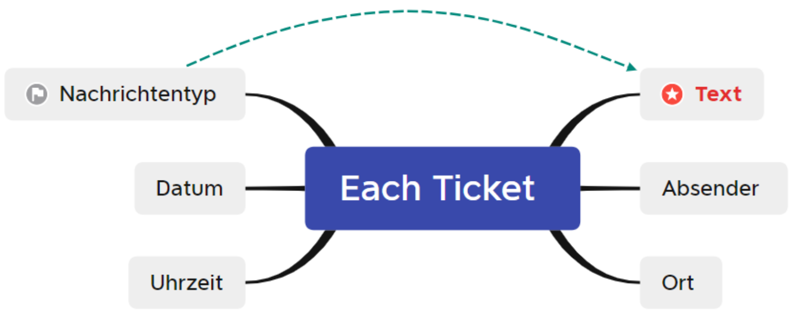


Figure 1.5: Attributes of each ticket

The most important information in each ticket is the text content sent by the users, including specific question information, staff's answers, etc. It is also the main data information that needs to be used in our project.

The “Nachrichtentyp” contained in each ticket is different. The input data that our clustering model needs to use is the problem descriptions sent by the users for the first time. The corresponding “Nachrichtentyp” is "Beschreibung" or "E-Mail von Kunde". We should filter out the data with specific problem descriptions for preprocessing.

In addition, the tickets sent by the users include different languages, so we should classify the tickets by using python libraries such as langdetect. We found that there are many languages such as English and German. In the subsequent preprocessing, attention should be paid to distinguishing data in different languages, such as extracting different keywords according to the type of language to construct a clustering model.

Furthermore, the tickets sent by the users may contain some illegible content and spelling errors. Therefore, corresponding preprocessing should be carried out before building the model, such as lower casing.

# 2 Team Introduction and Plan

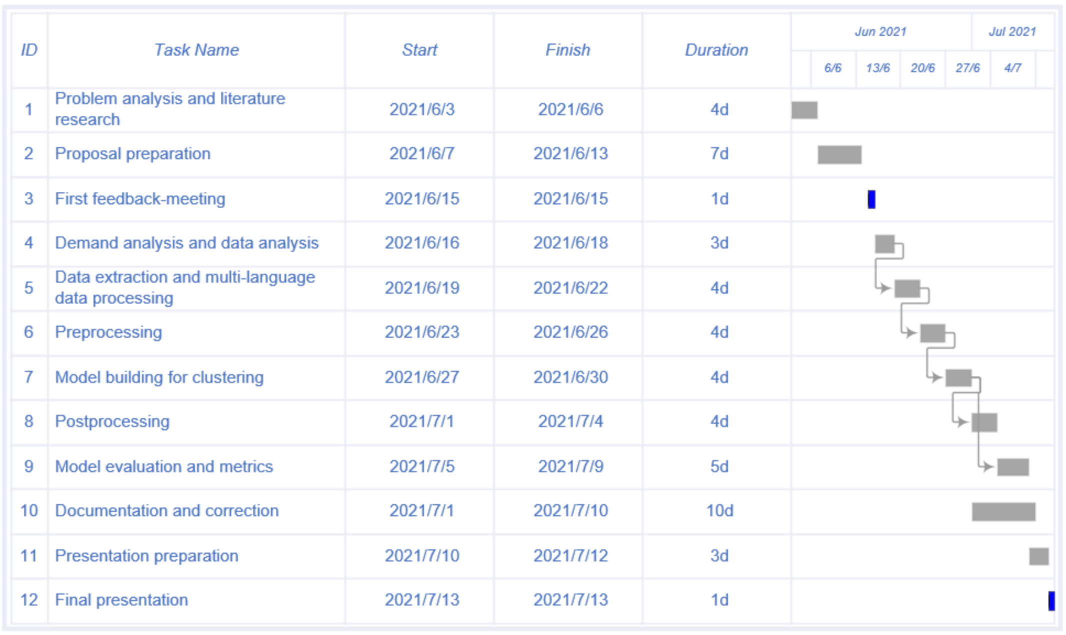
## 2.1 Team Members

Our team consisted of four people and each team member covered two main responsibility areas. The members of our group are Yujun Liu, Wenliang Peng, Chao Zhang and Yuanhao Zhong. The detailed overview of responsibilities is shown in the table below:

| Task | Name |
|---|---|
| Literature research and problem analysis | all |
| Demand analysis and data analysis | Peng, Zhong |
| Data extraction and multi-language data processing | Peng, Zhang |
| Preprocessing | Peng, Zhong |
| Model building for clustering | Liu, Zhang |
| Postprocessing | Liu, Zhang |
| Model evaluation and metrics | Liu, Zhong |
| Correction and documentation | all |

## 2.2 Project Plan

Gantt Chart Plan



| ID | Task Name | Start | Finish | Duration |
|---|---|---|---|---|
| 1 | Problem analysis and literature research | 2021/6/3 | 2021/6/6 | 4d |
| 2 | Proposal preparation | 2021/6/7 | 2021/6/13 | 7d |
| 3 | First feedback-meeting | 2021/6/15 | 2021/6/15 | 1d |
| 4 | Demand analysis and data analysis | 2021/6/16 | 2021/6/18 | 3d |
| 5 | Data extraction and multi-language data processing | 2021/6/19 | 2021/6/22 | 4d |
| 6 | Preprocessing | 2021/6/23 | 2021/6/26 | 4d |
| 7 | Model building for clustering | 2021/6/27 | 2021/6/30 | 4d |
| 8 | Postprocessing | 2021/7/1 | 2021/7/4 | 4d |
| 9 | Model evaluation and metrics | 2021/7/5 | 2021/7/9 | 5d |
| 10 | Documentation and correction | 2021/7/1 | 2021/7/10 | 10d |
| 11 | Presentation preparation | 2021/7/10 | 2021/7/12 | 3d |
| 12 | Final presentation | 2021/7/13 | 2021/7/13 | 1d |

# 3 Modelling Task

This part is the main modelling task for our project. The exact process is illustrated below:

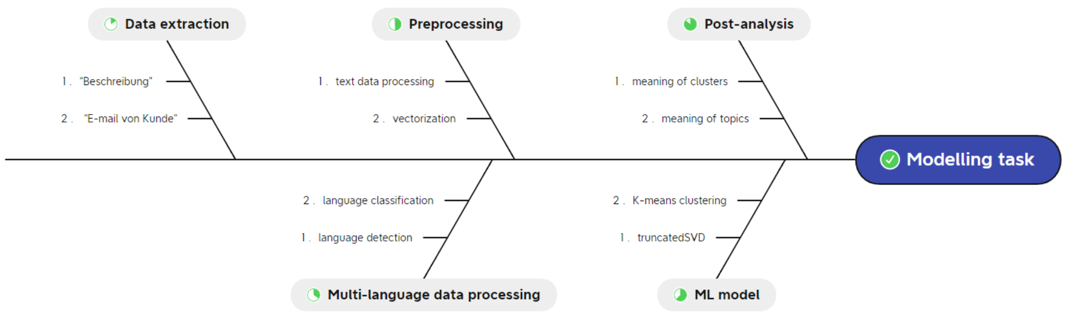
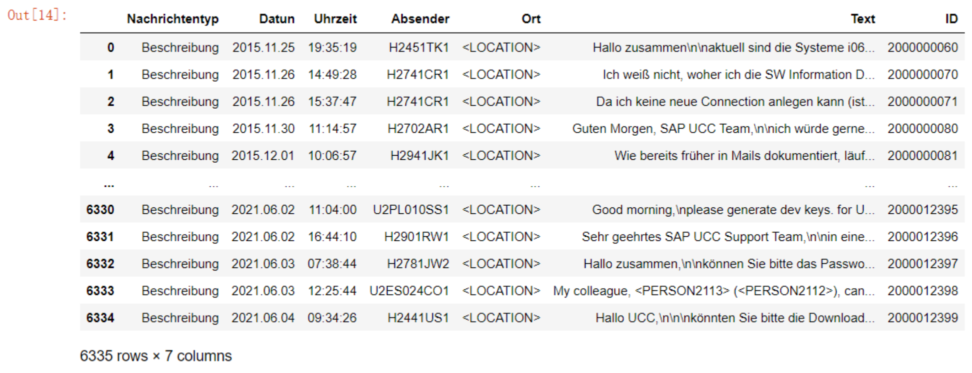


Figure 3.1: Flowchart of modelling work

## 3.1 Data Extraction

The original data we need to use is the descriptions of the problems sent by users for the first time, that is, the text content of the type "Beschreibung" and "E-mail von Kunde". Due to Alert tickets, blank tickets and tickets with missing problem descriptions, we first need to isolate all valid tickets. After loading all the tickets, we built a new data frame to read all valid tickets. Then after verification, a total of 6,335 tickets we need to use are obtained. By extracting texts of type "Beschreibung" and "E-mail von Kunde", we got the problem descriptions sent by valid users. The following figure shows some examples of valid tickets.



Out[14]:

| | Nachrichtentyp | Datun | Uhrzeit | Absender | Ort | Text | ID |
|---|---|---|---|---|---|---|---|
| **0** | Beschreibung | 2015.11.25 | 19:35:19 | H2451TK1 | <LOCATION> | Hallo zusammen\n\naktuell sind die Systeme i06... | 2000000060 |
| **1** | Beschreibung | 2015.11.26 | 14:49:28 | H2741CR1 | <LOCATION> | Ich weiß nicht, woher ich die SW Information D... | 2000000070 |
| **2** | Beschreibung | 2015.11.26 | 15:37:47 | H2741CR1 | <LOCATION> | Da ich keine neue Connection anlegen kann (ist... | 2000000071 |
| **3** | Beschreibung | 2015.11.30 | 11:14:57 | H2702AR1 | <LOCATION> | Guten Morgen, SAP UCC Team,\n\nich würde gerne... | 2000000080 |
| **4** | Beschreibung | 2015.12.01 | 10:06:57 | H2941JK1 | <LOCATION> | Wie bereits früher in Mails dokumentiert, läuf... | 2000000081 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **6330** | Beschreibung | 2021.06.02 | 11:04:00 | U2PL010SS1 | <LOCATION> | Good morning,\nplease generate dev keys. for U... | 2000012395 |
| **6331** | Beschreibung | 2021.06.02 | 16:44:10 | H2901RW1 | <LOCATION> | Sehr geehrtes SAP UCC Support Team,\n\nin eine... | 2000012396 |
| **6332** | Beschreibung | 2021.06.03 | 07:38:44 | H2781JW2 | <LOCATION> | Hallo zusammen,\n\nkönnen Sie bitte das Passwo... | 2000012397 |
| **6333** | Beschreibung | 2021.06.03 | 12:25:44 | U2ES024CO1 | <LOCATION> | My colleague, <PERSON2113> (<PERSON2112>), can... | 2000012398 |
| **6334** | Beschreibung | 2021.06.04 | 09:34:26 | H2441US1 | <LOCATION> | Hallo UCC,\n\n\nkönnten Sie bitte die Download... | 2000012399 |

6335 rows × 7 columns

Figure 3.2: Examples of valid tickets

## 3.2 Multi-language Processing

In this part, we are going to determine the language of each text in the data frame and also process tickets in different languages separately.

First, the languages need to be identified so that we can know how to handle these instances. We used the library langdetect to detect the language in the dataset. The results of the detection are shown below.

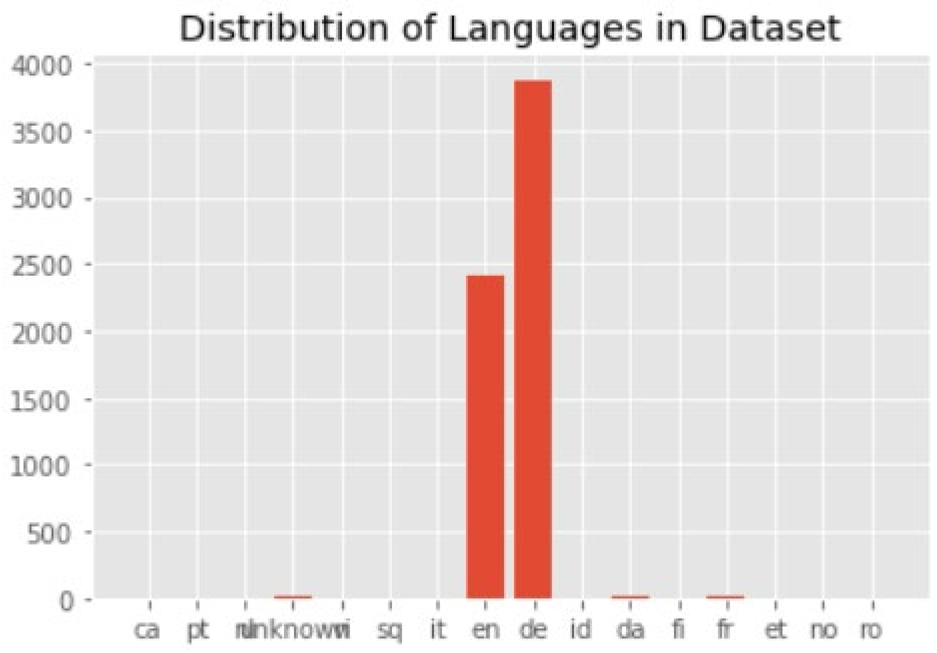
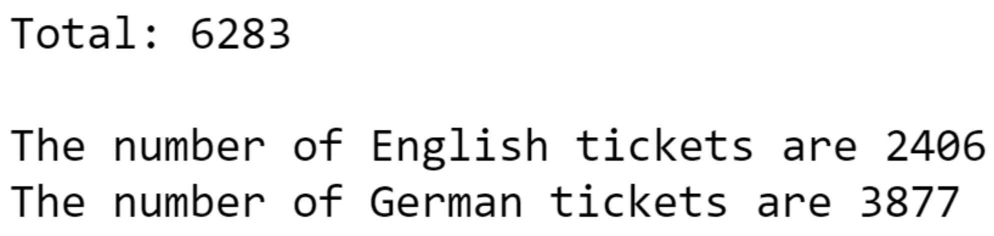


Figure 3.3: Language distribution of the tickets

As can be seen from the graph, the main languages in the dataset are English and German. Some other individual tickets were detected as some other languages, but after validation, we found that these were misclassified and they are still basically English or German. Therefore, we can consider mainly English and German, and only use tickets that are recognized as German and English text. If a subsequent user enters a ticket in another language, the system can simply forward the ticket to a member of staff for manual classification. The total number of tickets used and the corresponding languages are shown below:



```
Total: 6283

The number of English tickets are 2406
The number of German tickets are 3877
```

Figure 3.4: the number of tickets in German and English

## 3.3 Preprocessing

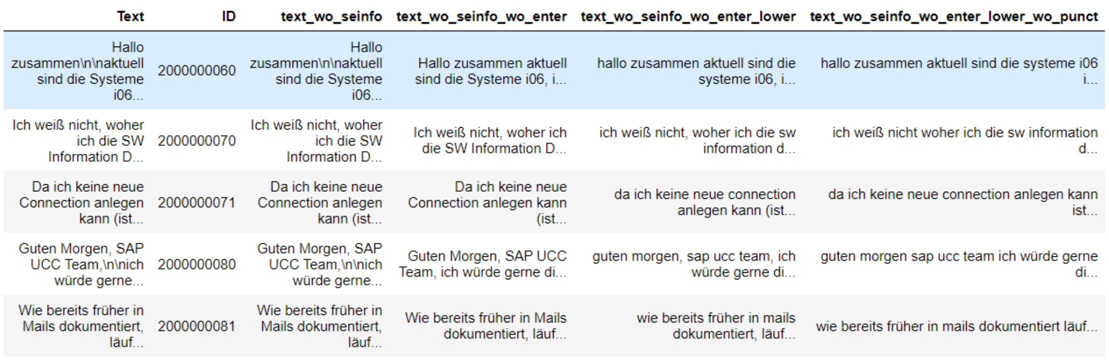
In this section, we performed preprocessing operations on the text data of all valid tickets (6283 in total). A series of common natural language processes were used to make the data form more uniform, remove useless distractions and facilitate subsequent vectorization. The preprocessing we used is as follows.

1) **Removal of sensitive words.** By removing the names of people and locations in the text, on the one hand, the interference of these unimportant words can be reduced, and on the other hand, the dimensionality of the text vector can be reduced.

2) **Removal of spaces and line feeds.** This step can remove useless information such as placeholder spaces.

3) **Lower casing.** Converting all words in the dataset to lower case is an effective way of removing redundancy. This is because words may appear several times in the text, some of them in lower case and others in upper case.

4) **Removal of punctuations.** This step can remove punctuation marks that have no real meaning. It is worth noting that the meaning of the words containing some punctuations may change if the punctuations are removed directly. For example, for the word “u-user”, if we directly remove the hyphen, it will turn into another wrong word “uuser”. So, we used spaces instead of punctuations to solve this problem.

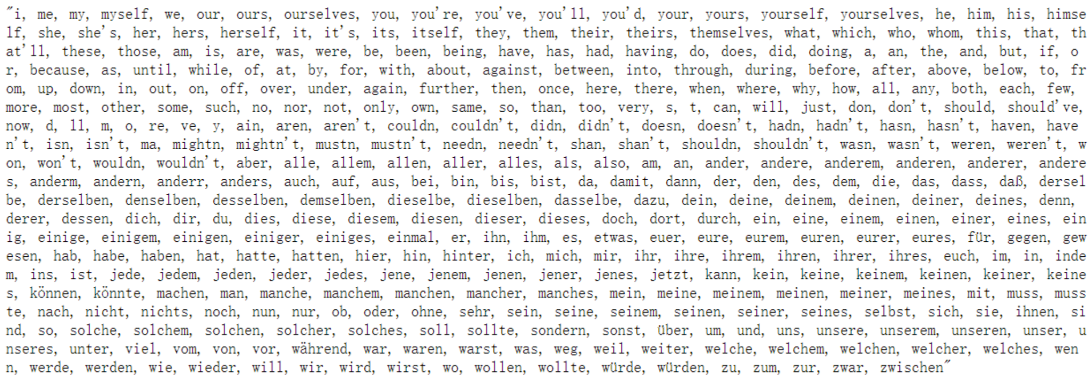


| Text | ID | text_wo_seinfo | text_wo_seinfo_wo_enter | text_wo_seinfo_wo_enter_lower | text_wo_seinfo_wo_enter_lower_wo_punct |
|---|---|---|---|---|---|
| Hallo zusammen\n\naktuell sind die Systeme i06... | 2000000060 | Hallo zusammen\n\naktuell sind die Systeme i06... | Hallo zusammen aktuell sind die Systeme i06, i... | hallo zusammen aktuell sind die systeme i06, i... | hallo zusammen aktuell sind die systeme i06 i... |
| Ich weiß nicht, woher ich die SW Information D... | 2000000070 | Ich weiß nicht, woher ich die SW Information D... | Ich weiß nicht, woher ich die SW Information D... | ich weiß nicht, woher ich die sw information d... | ich weiß nicht woher ich die sw information d... |
| Da ich keine neue Connection anlegen kann (ist... | 2000000071 | Da ich keine neue Connection anlegen kann (ist... | Da ich keine neue Connection anlegen kann (ist... | da ich keine neue connection anlegen kann (ist... | da ich keine neue connection anlegen kann ist... |
| Guten Morgen, SAP UCC Team,\n\nich würde gerne... | 2000000080 | Guten Morgen, SAP UCC Team,\n\nich würde gerne... | Guten Morgen, SAP UCC Team, ich würde gerne di... | guten morgen, sap ucc team, ich würde gerne di... | guten morgen sap ucc team ich würde gerne di... |
| Wie bereits früher in Mails dokumentiert, läuf... | 2000000081 | Wie bereits früher in Mails dokumentiert, läuf... | Wie bereits früher in Mails dokumentiert, läuf... | wie bereits früher in mails dokumentiert, läuf... | wie bereits früher in mails dokumentiert läuf... |

Figure 3.5: Examples of partial preprocessing

The figure above shows examples of the text data after the first 4 preprocessing steps. From left to right are: the row text, ticket ID, text after removing sensitive words, text after removing spaces and line feeds, text after lower casing, and text after removing punctuations.

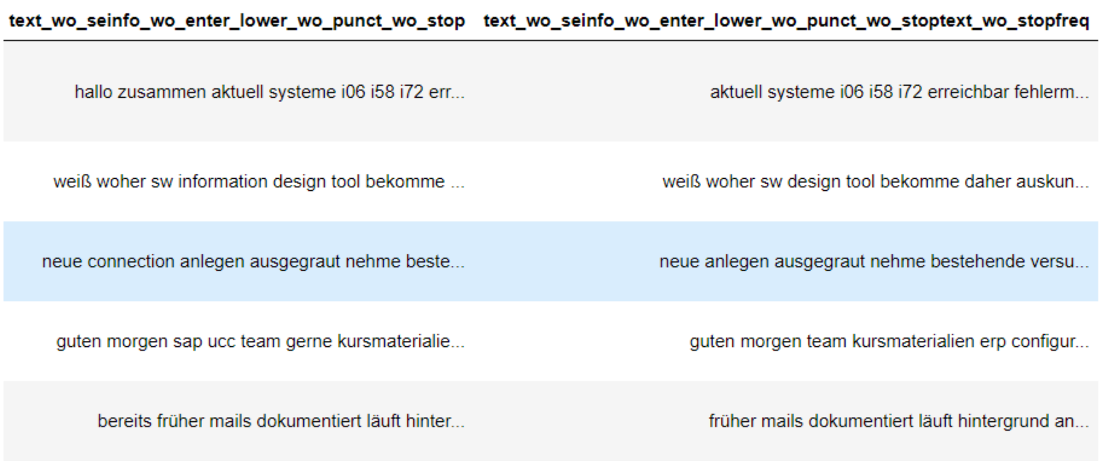
5) **Removal of stop-words**. Usually, text data contains a large number of stop-words. These words haven't any significant impact on the data, so we can remove them. The following figure shows the stop-words removed in English and German.



```
"i, me, my, myself, we, our, ours, ourselves, you, you're, you've, you'll, you'd, your, yours, yourself, yourselves, he, him, his, himse
lf, she, she's, her, hers, herself, it, it's, its, itself, they, them, their, theirs, themselves, what, which, who, whom, this, that, th
at'll, these, those, am, is, are, was, were, be, been, being, have, has, had, having, do, does, did, doing, a, an, the, and, but, if, o
r, because, as, until, while, of, at, by, for, with, about, against, between, into, through, during, before, after, above, below, to, fr
om, up, down, in, out, on, off, over, under, again, further, then, once, here, there, when, where, why, how, all, any, both, each, few,
more, most, other, some, such, no, nor, not, only, own, same, so, than, too, very, s, t, can, will, just, don, don't, should, should've,
now, d, ll, m, o, re, ve, y, ain, aren, aren't, couldn, couldn't, didn, didn't, doesn, doesn't, hadn, hadn't, hasn, hasn't, haven, have
n't, isn, isn't, ma, mightn, mightn't, mustn, mustn't, needn, needn't, shan, shan't, shouldn, shouldn't, wasn, wasn't, weren, weren't, w
on, won't, wouldn, wouldn't, aber, alle, allem, allen, aller, alles, als, also, am, an, ander, andere, anderem, anderen, anderer, andere
s, anderm, andern, anderr, anders, auch, auf, aus, bei, bin, bis, bist, da, damit, dann, der, den, des, dem, die, das, dass, daß, dersel
be, derselben, denselben, desselben, demselben, dieselbe, dieselben, dasselbe, dazu, dein, deine, deinem, deinen, deiner, deines, denn,
derer, dessen, dich, dir, du, dies, diese, diesem, diesen, dieser, dieses, doch, dort, durch, ein, eine, einem, einen, einer, eines, ein
ig, einige, einigem, einigen, einiger, einiges, einmal, er, ihn, ihm, es, etwas, euer, eure, eurem, euren, eurer, eures, für, gegen, gew
esen, hab, habe, haben, hat, hatte, hatten, hier, hin, hinter, ich, mich, mir, ihr, ihre, ihrem, ihren, ihrer, ihres, euch, im, in, inde
m, ins, ist, jede, jedem, jeden, jeder, jedes, jene, jenem, jenen, jener, jenes, jetzt, kann, kein, keine, keinem, keinen, keiner, keine
s, können, könnte, machen, man, manche, manchem, manchen, mancher, manches, mein, meine, meinem, meinen, meiner, meines, mit, muss, muss
te, nach, nicht, nichts, noch, nun, nur, ob, oder, ohne, sehr, sein, seine, seinem, seinen, seiner, seines, selbst, sich, sie, ihnen, si
nd, so, solche, solchem, solchen, solcher, solches, soll, sollte, sondern, sonst, über, um, und, uns, unsere, unserem, unseren, unser, u
nseres, unter, viel, vom, von, vor, während, war, waren, warst, was, weg, weil, weiter, welche, welchem, welchen, welcher, welches, wen
n, werde, werden, wie, wieder, will, wir, wird, wirst, wo, wollen, wollte, würde, würden, zu, zum, zur, zwar, zwischen"
```

Figure 3.6: Commonly used stop-words

6) **Removal of frequent words.** There are a lot of high-frequency words appearing in a lot of tickets, such as some greetings, which may cause some interference to our model. Therefore, we filtered out 130 highest-frequency words, manually screened out the words that have no practical meaning to the problem descriptions, and removed all of them.



| text_wo_seinfo_wo_enter_lower_wo_punct_wo_stop | text_wo_seinfo_wo_enter_lower_wo_punct_wo_stoptext_wo_stopfreq |
|---|---|
| hallo zusammen aktuell systeme i06 i58 i72 err... | aktuell systeme i06 i58 i72 erreichbar fehlerm... |
| weiß woher sw information design tool bekomme ... | weiß woher sw design tool bekomme daher auskun... |
| neue connection anlegen ausgegraut nehme beste... | neue anlegen ausgegraut nehme bestehende versu... |
| guten morgen sap ucc team gerne kursmaterialie... | guten morgen team kursmaterialien erp configur... |
| bereits früher mails dokumentiert läuft hinter... | früher mails dokumentiert läuft hintergrund an... |

Figure 3.7: Examples of partial preprocessing

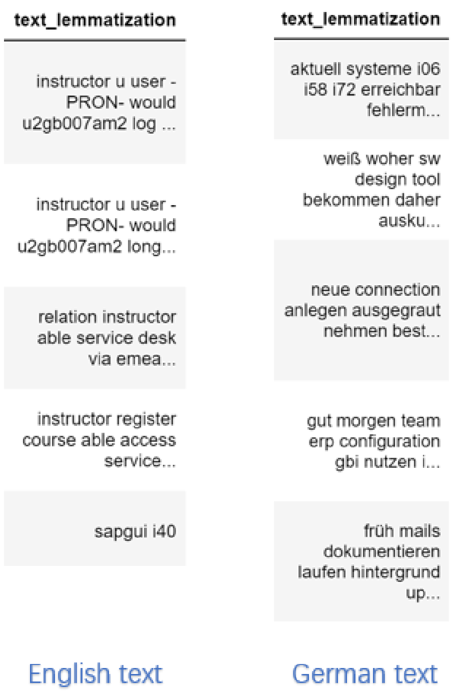
The figure above shows examples of the text data after the first 6 preprocessing steps: text after removing stop-words (left) and text after removing frequent words (right).

7) **Removal of Rare words.** In this step, we removed words that only appeared

once/twice. This greatly reduces the dimensionality of the text data and prepares for the subsequent vectorization.

8) **Removal of numbers and words starting with a number.** The text data also contains a lot of numbers that are not relevant to the content of problem descriptions, such as time information. So, we also should remove them.

9) **Lemmatization.** This is a very important step in preprocessing. In simple terms, lemmatization is the process of removing the affixes from a word and extracting the main part of the word. lemmatization is based on the dictionary and transforms the complex form of a word into its most basic form. Instead of simply removing the prefixes and suffixes, lemmatization transforms the word according to the dictionary. This method can help us to find valid text information quickly to a large extent. Some Examples of the text after lemmatization are shown in the following figure.



| text_lemmatization |
|---|
| instructor u user -PRON- would u2gb007am2 log ... |
| instructor u user -PRON- would u2gb007am2 long... |
| relation instructor able service desk via emea... |
| instructor register course able access service... |
| sapgui i40 |

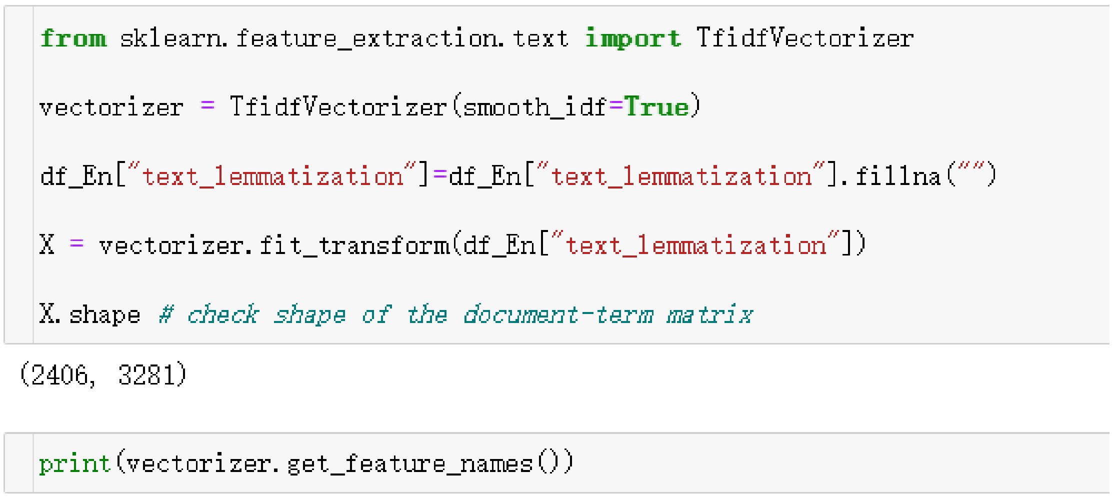
| text_lemmatization |
|---|
| aktuell systeme i06 i58 i72 erreichbar fehlerm... |
| weiß woher sw design tool bekommen daher ausku... |
| neue connection anlegen ausgegraut nehmen best... |
| gut morgen team erp configuration gbi nutzen i... |
| früh mails dokumentieren laufen hintergrund up... |

Figure 3.8: Examples after lemmatization

As a comparison, we also used the method of stemming. Stemming is the process of converting any word in the data to its root form. It is a process based on patterns in linguistic morphology that removes inflected or derived forms of affixes and unifies the different forms of a word into one representative standard form (stem). By comparing the results, we found that lemmatization was better at reproducing German words, so we chose to use lemmatization.

10) **Vectorization.** This step is to transform the preprocessed text data into vectors that can be used as input to the clustering model. We used the TF-IDF algorithm to implement the text vectorization. TF-IDF (term frequency-inverse document frequency) is a common weighting technique used in information retrieval and data mining. TF is

the term frequency and IDF is the inverse document frequency index TF is the Term Frequency and IDF is the Inverse Document Frequency. The algorithm uses the frequency of word occurrences as the feature of text data. To implement this algorithm, we use the “TfidfVectorizer” in the library scikit-learn to preprocess and vectorize for the algorithms. As shown in the figure below, we can call the “. get_feature_names ()" function to view the generated features.



```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(smooth_idf=True)

df_En["text_lemmatization"]=df_En["text_lemmatization"].fillna("")

X = vectorizer.fit_transform(df_En["text_lemmatization"])

X.shape # check shape of the document-term matrix
```

```
(2406, 3281)
```

```
print(vectorizer.get_feature_names())
```

Figure 3.9: TF-IDF vectorization

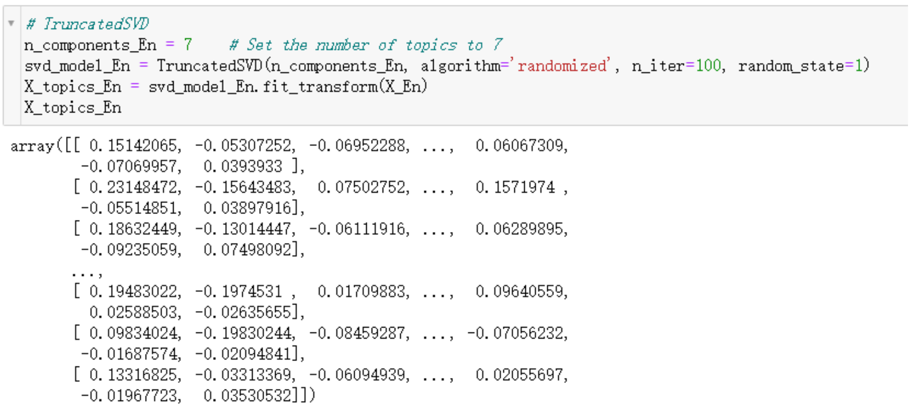
## 3.4 ML Model

### 3.4.1 Truncated SVD

After preprocessing, we got a vector X, which stores the text information of all tickets. We now want to apply a clustering algorithm to this vector X to study the topic focusing on the text information of all ticket samples.

We first tried the method of latent semantic analysis (LSA). Through topic analysis by LSA, we can get the key topic in the corpus, which means the importance of each word in the topic, and the tendency of each ticket sample on each topic. And based on them, keywords and representative texts corresponding to the theme can be obtained. LSA is just a truncated singular value decomposition of a (very high-rank and sparse) document-term matrix, only retaining the largest singular value of n_components. The process of LSA is as follows:

Using Truncated SVD, transform the original feature vector X with a scale of N*D (number of ticket’s samples, number of words) into a new feature vector X_topics with a scale of N*T (number of ticket’s samples, number of topics):

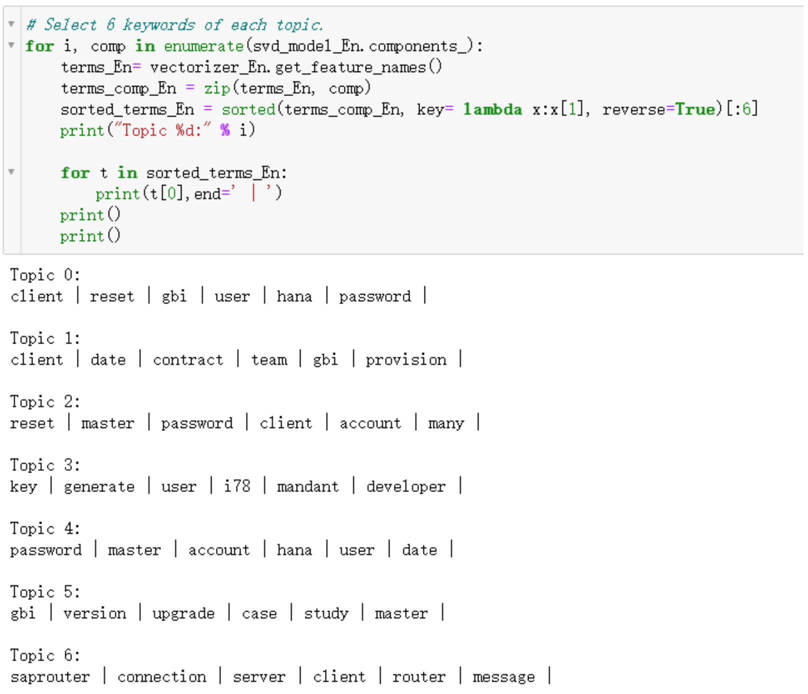


```
# TruncatedSVD
n_components_En = 7    # Set the number of topics to 7
svd_model_En = TruncatedSVD(n_components_En, algorithm='randomized', n_iter=100, random_state=1)
X_topics_En = svd_model_En.fit_transform(X_En)
X_topics_En
```

```
array([[ 0.15142065, -0.05307252, -0.06952288, ...,  0.06067309,
        -0.07069957,  0.0393933 ],
       [ 0.23148472, -0.15643483,  0.07502752, ...,  0.1571974 ,
        -0.05514851,  0.03897916],
       [ 0.18632449, -0.13014447, -0.06111916, ...,  0.06289895,
        -0.09235059,  0.07498092],
       ...,
       [ 0.19483022, -0.1974531 ,  0.01709883, ...,  0.09640559,
         0.02588503, -0.02635655],
       [ 0.09834024, -0.19830244, -0.08459287, ..., -0.07056232,
        -0.01687574, -0.02094841],
       [ 0.13316825, -0.03313369, -0.06094939, ...,  0.02055697,
        -0.01967723,  0.03530532]])
```

Figure 3.10: Truncated SVD Algorithm

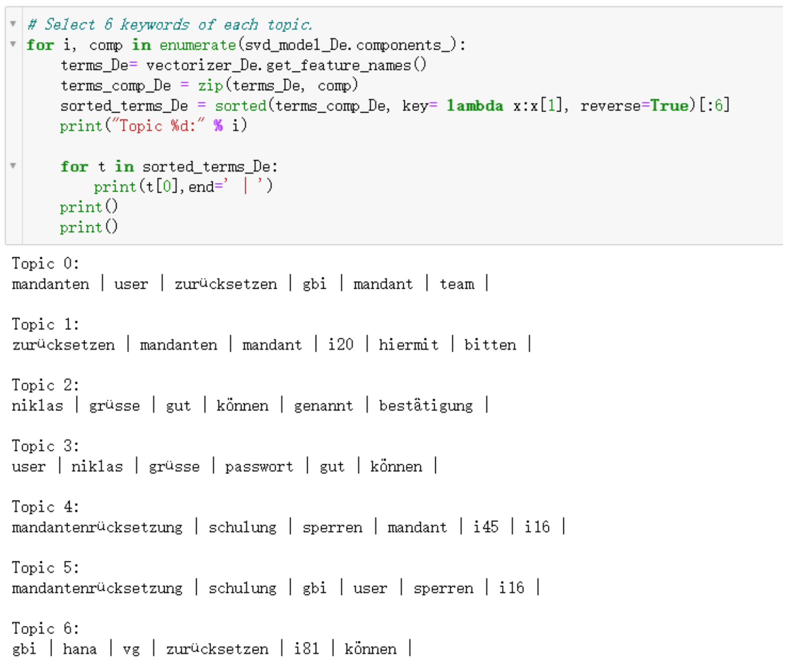
Svd_model.components_ is a vector with a scale of T*D (number of topics, number of words), and the elements in (i, j) represent the weight of word j on topic i. The higher the weight of word j, it can be considered more representative on the topic I, and we use this to select the most representative keywords of the topics.



```
# Select 6 keywords of each topic.
for i, comp in enumerate(svd_model_En.components_):
    terms_En= vectorizer_En.get_feature_names()
    terms_comp_En = zip(terms_En, comp)
    sorted_terms_En = sorted(terms_comp_En, key= lambda x:x[1], reverse=True)[:6]
    print("Topic %d:" % i)

    for t in sorted_terms_En:
        print(t[0],end=' | ')
    print()
    print()
```

```
Topic 0:
client | reset | gbi | user | hana | password |

Topic 1:
client | date | contract | team | gbi | provision |

Topic 2:
reset | master | password | client | account | many |

Topic 3:
key | generate | user | i78 | mandant | developer |

Topic 4:
password | master | account | hana | user | date |

Topic 5:
gbi | version | upgrade | case | study | master |

Topic 6:
saprouter | connection | server | client | router | message |
```

Figure 3.11: Topics after Truncated SVD Algorithm (English)



```python
# Select 6 keywords of each topic.
for i, comp in enumerate(svd_model_De.components_):
    terms_De= vectorizer_De.get_feature_names()
    terms_comp_De = zip(terms_De, comp)
    sorted_terms_De = sorted(terms_comp_De, key= lambda x:x[1], reverse=True)[:6]
    print("Topic %d:" % i)

    for t in sorted_terms_De:
        print(t[0],end=' | ')
    print()
    print()
```

```
Topic 0:
mandanten | user | zurücksetzen | gbi | mandant | team |

Topic 1:
zurücksetzen | mandanten | mandant | i20 | hiermit | bitten |

Topic 2:
niklas | grüsse | gut | können | genannt | bestätigung |

Topic 3:
user | niklas | grüsse | passwort | gut | können |

Topic 4:
mandantenrücksetzung | schulung | sperren | mandant | i45 | i16 |

Topic 5:
mandantenrücksetzung | schulung | gbi | user | sperren | i16 |

Topic 6:
gbi | hana | vg | zurücksetzen | i81 | können |
```

Figure 3.12: Topics after Truncated SVD Algorithm (German)

Although we got the topics using LSA, we did not find a suitable method to match the topics with the ticket's samples, so we tried to use K-means for clustering.

### 3.4.2 K-means Clustering

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms. The working principle of the algorithm is as follows:

- First, we randomly initialize k points, which are called means.
- We classify each item as its closest mean and update the coordinates of the mean, which is the average value of items classified by that mean so far.
- We repeat this process for a given number of iterations, and finally, we get the clusters.

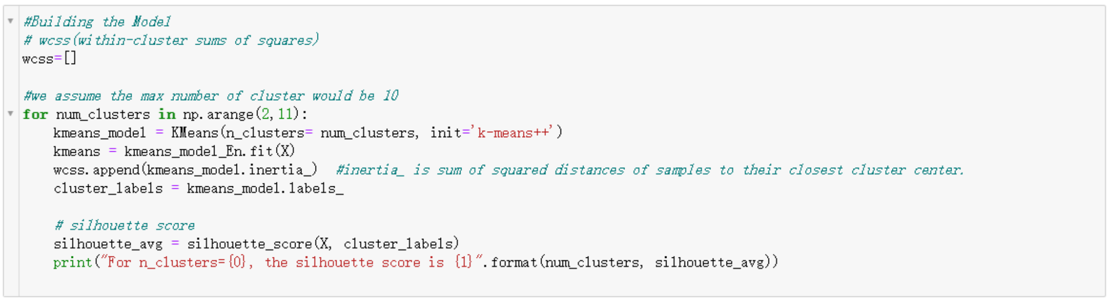
A very important part of K-means is to find the best number of clusters. The basic step of any unsupervised algorithm is to determine the optimal number of clusters to which the data can be clustered. The Elbow method and Silhouette Analysis are the popular methods to determine the optimal value of k.

The elbow curve is similar to a human elbow, and the value of k corresponding to the "elbow joint" part is the most appropriate k, but sometimes the "elbow joint" of the elbow curve is not obvious, so we can use the silhouette score to analyze, The the silhouette score is a criterion for judging whether the clustering is good or bad, and it is calculated by combining two indexes of the degree of aggregation within a cluster and the degree of separation between clusters.

Silhouette score can be calculated with (p-q) / max(p,q), where p is the average distance to the point in the closest cluster to which the data point does not belong, q is the average intra-cluster distance to all points in its cluster.

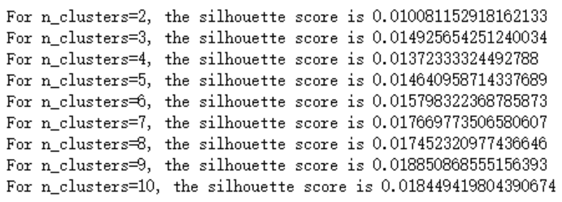
- The value of the silhouette score range is between -1 to 1.
- A score close to 1 indicates that the data point is very similar to other data points in the cluster.
- A score close to -1 indicates that the data point is not similar to the data points in its cluster.

The following is the code and output result of using the elbow method and silhouette analysis to determine the optimal number of clusters:



```
#Building the Model
# wcss(within-cluster sums of squares)
wcss=[]

#we assume the max number of cluster would be 10
for num_clusters in np.arange(2,11):
    kmeans_model = KMeans(n_clusters= num_clusters, init='k-means++')
    kmeans = kmeans_model_En.fit(X)
    wcss.append(kmeans_model.inertia_)  #inertia_ is sum of squared distances of samples to their closest cluster center.
    cluster_labels = kmeans_model.labels_

    # silhouette score
    silhouette_avg = silhouette_score(X, cluster_labels)
    print("For n_clusters={0}, the silhouette score is {1}".format(num_clusters, silhouette_avg))
```

Figure 3.13: The method of finding the best number of clusters



```
For n_clusters=2, the silhouette score is 0.010081152918162133
For n_clusters=3, the silhouette score is 0.014925654251240034
For n_clusters=4, the silhouette score is 0.013723333244492788
For n_clusters=5, the silhouette score is 0.014640958714337689
For n_clusters=6, the silhouette score is 0.015798322368785873
For n_clusters=7, the silhouette score is 0.017669773506580607
For n_clusters=8, the silhouette score is 0.017452320977436646
For n_clusters=9, the silhouette score is 0.018850868555156393
For n_clusters=10, the silhouette score is 0.018449419804390674
```

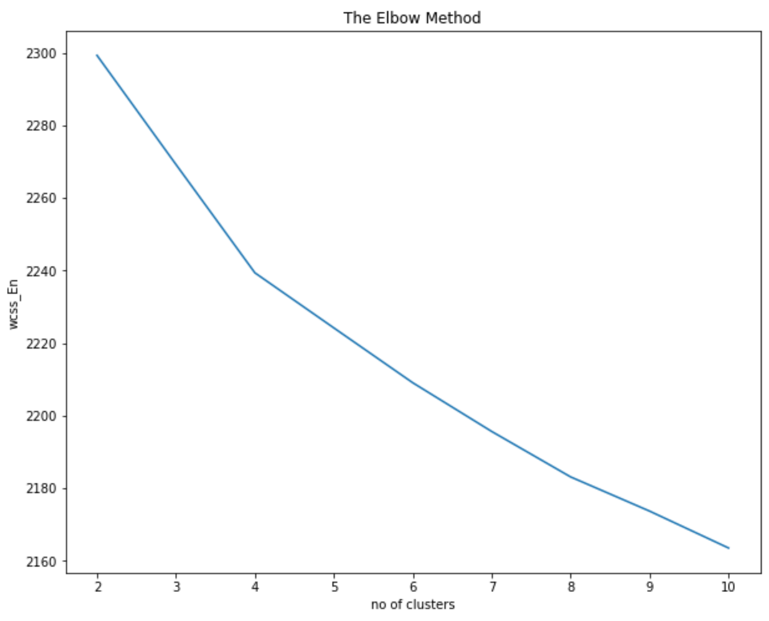


Figure 3.14: Results of elbow method and silhouette analysis (English)

```
For n_clusters=2, the silhouette score is 0.0049440125360490006
For n_clusters=3, the silhouette score is 0.005295503895894618
For n_clusters=4, the silhouette score is 0.005598898154542968
For n_clusters=5, the silhouette score is 0.006730486566390354
For n_clusters=6, the silhouette score is 0.007100384809189779
For n_clusters=7, the silhouette score is 0.008774731048306684
For n_clusters=8, the silhouette score is 0.00903907090425511
For n_clusters=9, the silhouette score is 0.010703021913361066
For n_clusters=10, the silhouette score is 0.01113657411551829
```

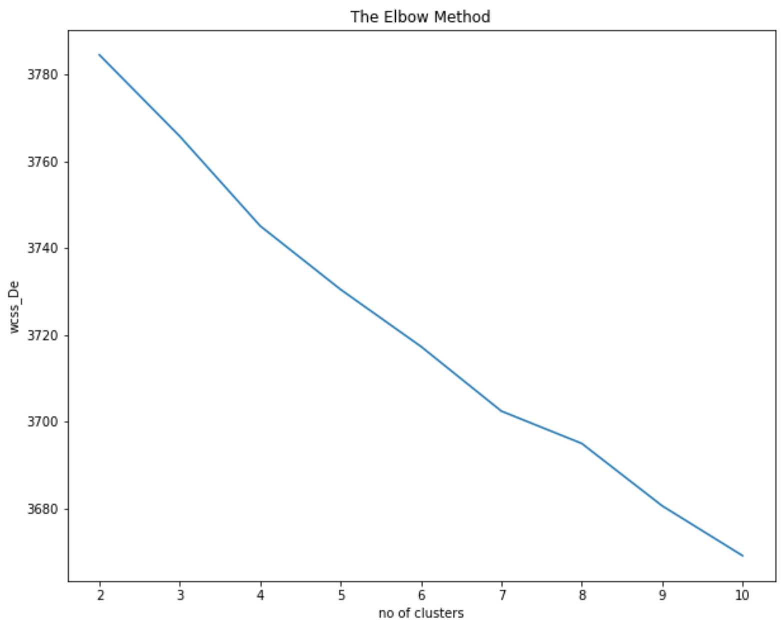


Figure 3.15: Results of elbow method and silhouette analysis (German)

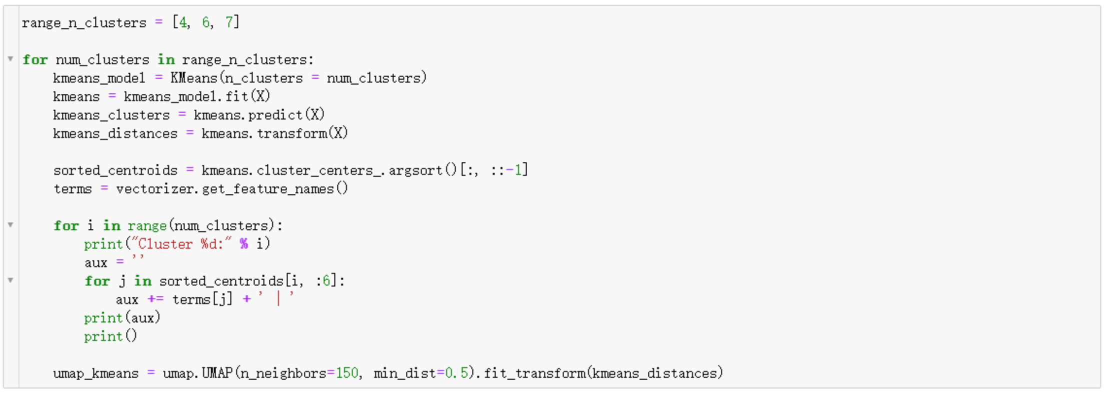
We choose the number of clusters to be 4, 6, and 7 to visualize the data.

As we mentioned before, the feature vector X we got after TfidfVectorizer processing is high-dimensional, and we need to visualize the cluster distance space obtained after K-means clustering on the 2D plane.

Uniform Manifold Approximation and Projection (UMAP) is a brand-new dimensionality reduction technology, especially suitable for the visualization of high-dimensional data sets. Its goal is to find a representation of high-dimensional points in a low-dimensional space (usually a 2D plane). In terms of visual quality, compared with t-SNE, UMAP retains a more global structure and has superior operating performance.

Similar to the svd_model.components_ we referred to before, kmeans.cluster_centers_ is also a vector with a scale of N*D (number of clusters, number of words), and the elements in (i, j) represent the weight of word j on cluster i.  The higher the weight of word j, the more representative it is on cluster i, and we use this to select the most representative keywords of clusters.

The following figure is an example of when the number of clusters is selected to be 4, output 6 keywords for each cluster, and visualize the data points of all tickets' samples.



```
range_n_clusters = [4, 6, 7]

for num_clusters in range_n_clusters:
    kmeans_model = KMeans(n_clusters = num_clusters)
    kmeans = kmeans_model.fit(X)
    kmeans_clusters = kmeans.predict(X)
    kmeans_distances = kmeans.transform(X)

    sorted_centroids = kmeans.cluster_centers_.argsort()[:, ::-1]
    terms = vectorizer.get_feature_names()

    for i in range(num_clusters):
        print("Cluster %d:" % i)
        aux = ''
        for j in sorted_centroids[i, :6]:
            aux += terms[j] + ' | '
        print(aux)
        print()

    umap_kmeans = umap.UMAP(n_neighbors=150, min_dist=0.5).fit_transform(kmeans_distances)
```

Figure 3.16: UMAP method

```
Cluster 0:
student | client | gbi | access | user | hana |

Cluster 1:
reset | client | password | master | user | account |

Cluster 2:
key | generate | user | developer | i78 | mandant |

Cluster 3:
client | date | contract | team | university | provision |
```

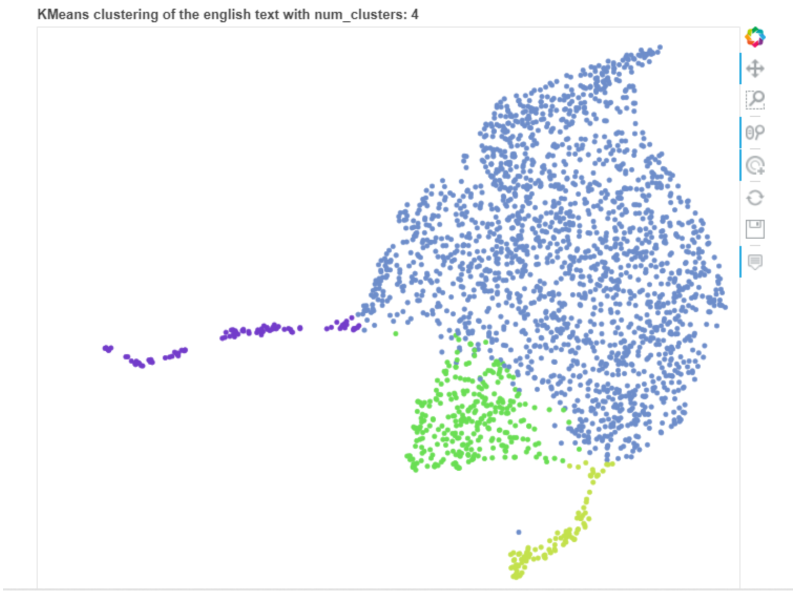
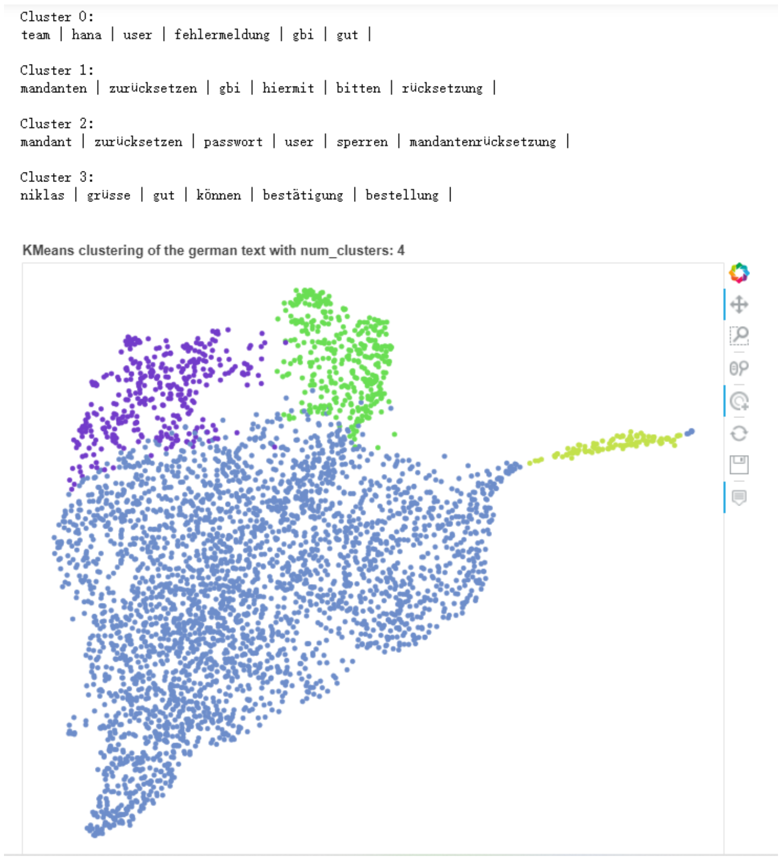


Figure 3.17: Result after clustering (English)

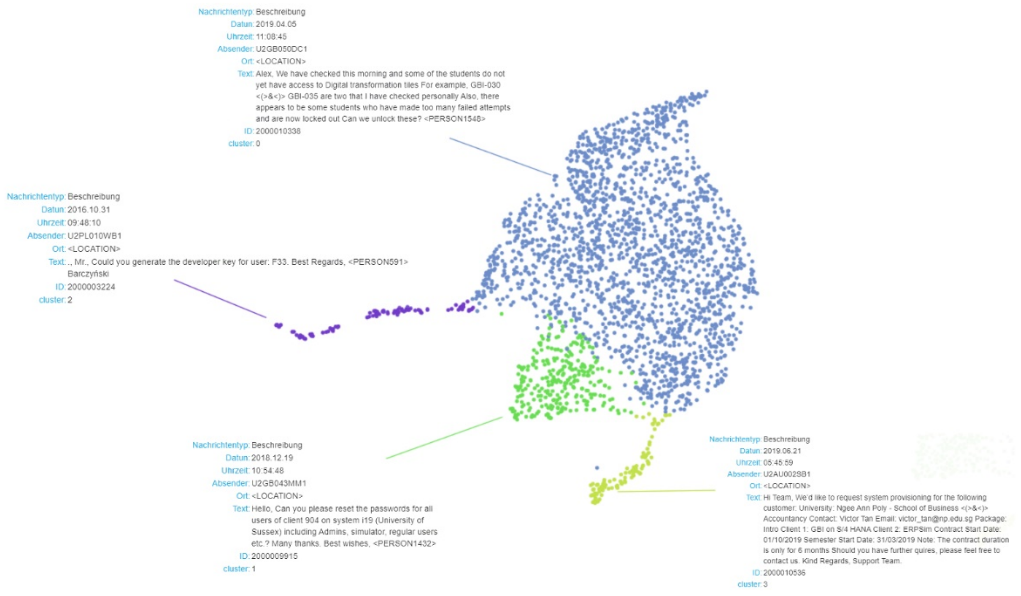


```
Cluster 0:
team | hana | user | fehlermeldung | gbi | gut |

Cluster 1:
mandanten | zurücksetzen | gbi | hiermit | bitten | rücksetzung |

Cluster 2:
mandant | zurücksetzen | passwort | user | sperren | mandantenrücksetzung |

Cluster 3:
niklas | grüsse | gut | können | bestätigung | bestellung |
```

Figure 3.18: Result after clustering (German)

When we click on one of the tickets' samples with the mouse, we can get some information about this ticket's sample, such as information type, time, ID, etc. The most important is the text information, as shown in the figure, we have selected a sample from each cluster as a display. By looking at the keywords in the cluster corresponding to the sample we can see that the textual information of the sample corresponds roughly to the keywords of the cluster to which the sample belongs.

In addition, we found that English Cluster3 contains a lot of sensitive information. Almost all of the names of people, schools, and contact details are not treated for sensitive words. Therefore, the sensitive word processing in the ticket support system can be paid more attention to the data in this cluster.



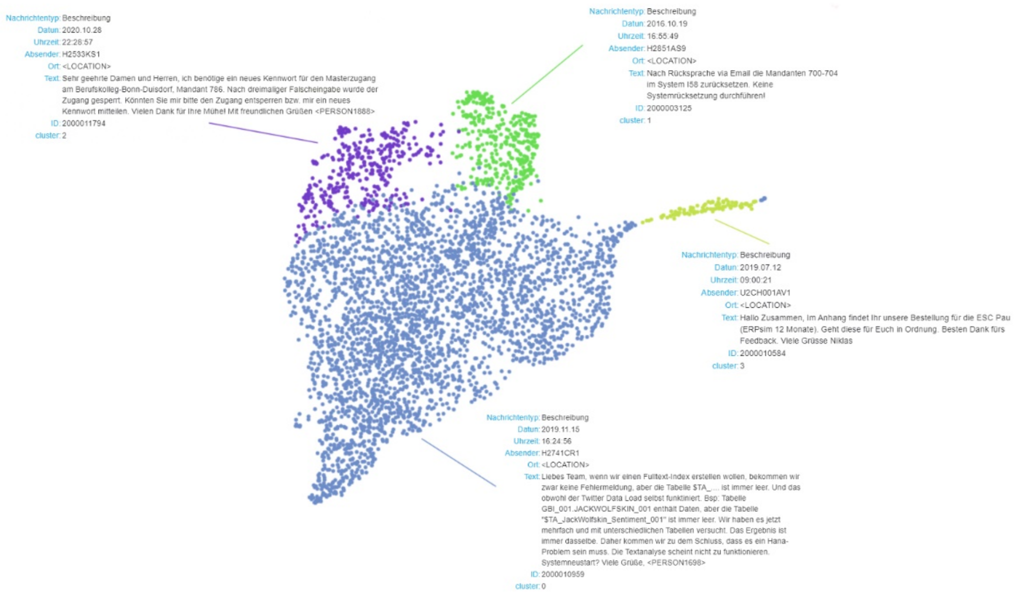
Nachrichtentyp: Beschreibung
Datun: 2019.04.05
Uhrzeit: 11:08:45
Absender: U2GB050DC1
Ort: <LOCATION>
Text: Alex, We have checked this morning and some of the students do not yet have access to Digital transformation tiles For example, GBI-030 <(>&<)> GBI-035 are two that I have checked personally Also, there appears to be some students who have made too many failed attempts and are now locked out Can we unlock these? <PERSON1548>
ID: 2000010338
cluster: 0

Nachrichtentyp: Beschreibung
Datun: 2016.10.31
Uhrzeit: 09:48:10
Absender: U2PL010WB1
Ort: <LOCATION>
Text: ., Mr., Could you generate the developer key for user: F33. Best Regards, <PERSON591> Barczyński
ID: 2000003224
cluster: 2

Nachrichtentyp: Beschreibung
Datun: 2018.12.19
Uhrzeit: 10:54:48
Absender: U2GB043MM1
Ort: <LOCATION>
Text: Hello, Can you please reset the passwords for all users of client 904 on system i19 (University of Sussex) including Admins, simulator, regular users etc.? Many thanks. Best wishes, <PERSON1432>
ID: 2000009915
cluster: 1

Nachrichtentyp: Beschreibung
Datun: 2019.06.21
Uhrzeit: 05:45:59
Absender: U2AU002SB1
Ort: <LOCATION>
Text: Hi Team, We'd like to request system provisioning for the following customer: University: Ngee Ann Poly - School of Business <(>&<)> Accountancy Contact: Victor Tan Email: victor_tan@np.edu.sg Package: Intro Client 1: GBI on S/4 HANA Client 2: ERPSim Contract Start Date: 01/10/2019 Semester Start Date: 31/03/2019 Note: The contract duration is only for 6 months Should you have further quires, please feel free to contact us. Kind Regards, Support Team.
ID: 2000010536
cluster: 3

Figure 3.19: Clustering visualization of English data



Nachrichtentyp: Beschreibung
Datun: 2020.10.28
Uhrzeit: 22:28:57
Absender: H2533KS1
Ort: <LOCATION>
Text: Sehr geehrte Damen und Herren, ich benötige ein neues Kennwort für den Masterzugang am Berufskolleg-Bonn-Duisdorf, Mandant 786. Nach dreimaliger Falscheingabe wurde der Zugang gesperrt. Könnten Sie mir bitte den Zugang entsperren bzw. mir ein neues Kennwort mitteilen. Vielen Dank für Ihre Mühe! Mit freundlichen Grüßen <PERSON1888>
ID: 2000011794
cluster: 2

Nachrichtentyp: Beschreibung
Datun: 2016.10.19
Uhrzeit: 16:55:49
Absender: H2851AS9
Ort: <LOCATION>
Text: Nach Rücksprache via Email die Mandanten 700-704 im System I58 zurücksetzen. Keine Systemrücksetzung durchführen!
ID: 2000003125
cluster: 1

Nachrichtentyp: Beschreibung
Datun: 2019.07.12
Uhrzeit: 09:00:21
Absender: U2CH001AV1
Ort: <LOCATION>
Text: Hallo Zusammen, Im Anhang findet Ihr unsere Bestellung für die ESC Pau (ERPsim 12 Monate). Geht diese für Euch in Ordnung. Besten Dank fürs Feedback. Viele Grüsse Niklas
ID: 2000010584
cluster: 3

Nachrichtentyp: Beschreibung
Datun: 2019.11.15
Uhrzeit: 16:24:56
Absender: H2741CR1
Ort: <LOCATION>
Text: Liebes Team, wenn wir einen Fulltext-Index erstellen wollen, bekommen wir zwar keine Fehlermeldung, aber die Tabelle $TA_.... ist immer leer. Und das obwohl der Twitter Data Load selbst funktiniert. Bsp: Tabelle GBI_001.JACKWOLFSKIN_001 enthält Daten, aber die Tabelle "$TA_JackWolfskin_Sentiment_001" ist immer leer. Wir haben es jetzt mehrfach und mit unterschiedlichen Tabellen versucht. Das Ergebnis ist immer dasselbe. Daher kommen wir zu dem Schluss, dass es ein Hana-Problem sein muss. Die Textanalyse scheint nicht zu funktionieren. Systemneustart? Viele Grüße, <PERSON1698>
ID: 2000010959
cluster: 0

Figure 3.20: Clustering visualization of German data

## 3.5 Post-analysis

Through the establishment of the models, we obtained the topics that are most relevant to the descriptions of the problems entered by all users and the clusters according to the current dataset. In this step, we will analyze topics and clusters separately.

### 3.5.1 Meaning of Topics after Truncated SVD

Through the truncated SVD process, we downscaled the high-dimensional text data to the most important 7 dimensions/feature space and printed out the first 6 keywords of the 7 topics, the results are shown in the table below.

| NO. of topic | Keywords (English) | Keywords (German) |
|---|---|---|
| 1 | client \| reset \| gbi \| user \| hana \| password | mandanten \| user \| zurücksetzen \| gbi \| mandant \| team |
| 2 | client \| date \| contract \| team \| gbi \| provision | zurücksetzen \| mandanten \| mandant \| i20 \| hiermit \| bitten |
| 3 | reset \| master \| password \| client \| account \| many | niklas \| grüsse \| gut \| können \| genannt \| bestätigung |
| 4 | key \| generate \| user \| i78 \| mandant \| developer | user \| niklas \| grüsse \| passwort \| gut \| können |
| 5 | password \| master \| account \| hana \| user \| date | mandantenrücksetzung \| schulung \| sperren \| mandant \| i45 \| i16 |
| 6 | gbi \| version \| upgrade \| case \| study \| master | mandantenrücksetzung \| schulung \| gbi \| user \| sperren \| i16 |
| 7 | saprouter \| connection \| server \| client \| router \| message | gbi \| hana \| vg \| zurücksetzen \| i81 \| können |

Table 3.1: Keywords of the topics

(The German and English topics in the table do not correspond to each other.)

From the above table, we can see that our users ask more questions about password reset, key development, gbi upgrade in both English and German. By comparing the keywords in the English and German topics, we found that the German topics had more meaningless words such as bitten, grüsse, etc. The English topics are more meaningful. These topics give us a general idea of the common questions asked by the users. We can also improve the FAQs by including more frequently asked questions in the FAQs based on these topics. For users, this will improve the efficiency of solving common problems. For the company, these topics are a summary of the user's problems and can be used as a part of the overview of the data.

### 3.5.2 Meaning of Clusters after K-means Clustering

Through the K-means clustering method, based on the principle of the minimum sum of Euclidean distance, we clustered the preprocessed text data into 7 clusters. As shown in the figure, it is the first 6 keywords of the 7 clusters of English data and German data.

| NO. of clusters | Keywords (English) | Keywords (German) |
|---|---|---|
| 1 | key \| generate \| user \| i78 \| developer \| mandant \| | mandantenrücksetzung \| schulung \| sperren \| vertragsende \| i45 \| i16 \| |
| 2 | gbi \| version \| upgrade \| client \| access \| update \| | gui \| domain \| adresse \| zugreifen \| remote \| folgend \| |
| 3 | password \| master \| reset \| user \| a ccount \| lock \| | hana \| team \| vg \| können \| fehlermeldung \| gut \| |
| 4 | student \| case \| create \| study \| user \| account \| | mandanten \| gbi \| zurück \| rücksetzung \| setzen \| mandant \| |
| 5 | client \| date \| contract \| team \| university \| provision \| | niklas \| grüsse \| gut \| können \| bestätigung \| bestellung \| |
| 6 | reset \| client \| refresh \| many \| erpsim \| i45 \| | zurücksetzen \| mandanten \| mandant \| bitten \| i20 \| passwort \| |
| 7 | access \| hana \| client \| domain \| university \| team \| | user \| master \| passwort \| anlegen \| team \| entwicklerschlüssel \| |

Table 3.2: Keywords of the clusters

(The German and English topics in the table do not correspond to each other.)

Through the comparison of keywords in English and German clusters, similarly, we find that German clusters have more meaningless words, and the clustering of the English model is more meaningful. Therefore, for English data, we have obtained the significance of 7 clusters in practical applications through analysis:

| NO. of clusters | Keywords (English) | Meaning |
|---|---|---|
| 1 | key \| generate \| user \| i78 \| developer \| mandant \| | Key generation or developer |
| 2 | gbi \| version \| upgrade \| client \| access \| update \| | GBI version upgrade |

| | | |
|---|---|---|
| 3 | password \| master \| reset \| user \| a ccount \| lock \| | Password/Account reset |
| 4 | student \| case \| create \| study \| user \| account \| | Questions/Case study from students |
| 5 | client \| date \| contract \| team \| university \| provision \| | Questions about contracts |
| 6 | reset \| client \| refresh \| many \| erpsim \| i45 \| | ERPsim relevant |
| 7 | access \| hana \| client \| domain \| university \| team \| | Questions about HANA |

Table 3.3: Keywords and meaning of the English clusters

As can be seen from the table above, the clusters obtained by the K-means algorithm are meaningful for users' problems classification. For example, for the cluster of Password/Account reset, the keywords given by the K-means algorithm include master as well as lock, which can give some information about the reason why a ticket belongs to the cluster of Password/Account reset.

# 4 Evaluation

In this section, we evaluated our model as well as the project from two perspectives. By analyzing, we gained a deeper understanding of the model as well as the project.

## 4.1 Evaluation of the Model

In our previous modelling work, we modelled the English and German dataset separately to obtain separate clustering models for German- and English-speaking users. By comparing the German and English models we found:

1) In the loss function graph, when finding the best number of clusters by the elbow rule, the elbow of the English model is more pronounced, which allows us to find the number of clusters with better performance.
2) The English clustering model has a better Silhouette score in terms of the performance of the clustering results.
3) In the English model, the keywords in each topic and cluster are more representative and meaningful.

The results of the above comparison indicate that the English model has a better performance compared to the German model. Possible reasons for this are:

1) The libraries used for preprocessing are more suitable for English data.
2) German itself is more difficult to preprocess. For example, for German conjugation, separable verbs can be not handled well, resulting in many stems being read incompletely or incorrectly.

## 4.2 Evaluation of the Project

After completing all the modelling work, and inspired by the SWOT assessment methodology, we carried out a comprehensive and systematic evaluation of the entire project. SWOT analysis (or SWOT matrix) is a strategic planning technique used to help us identify strengths, weaknesses, opportunities, and threats related to project planning.

The strengths of our project are:

1) The project provides the most frequent topics by users, which gives a better understanding of the data entered by users, and can also provide a reference for the setting of FAQs.
2) The user's problem descriptions are broadly clustered and can be used as part of the preprocessing for the subsequent classification task.
3) The models don't rely on data from the ticket support system forms that may contain incorrect information but use the initial text information directly, which can ensure the correctness of the model.
4) The project provides a very clear process and approach to modelling using text data, and the analysis is very intuitive and can provide a basis and reference for subsequent modelling.

The weaknesses of our project are:

1) The project is not directly user-oriented and can only be used as a small early part of the ticket support system. Therefore, it doesn't have sufficient product value.
2) In terms of the model results, the keywords of some topics and clusters obtained are not clear, and some keywords have no practical meaning, such as bitten, grüsse, etc.
3) The preprocessing process doesn't work as expected: some irrelevant information cannot be completely removed, and some words cannot be restored, especially for the German data.

The opportunities and development prospects for our project are diverse. There are many directions for optimization and subsequent application of the project. The details will be carefully presented and discussed in the next section.

The threats of our project are:

1) Inaccurate keywords of topics as well as clusters can have a direct negative impact on

subsequent applications.

2) In contrast to the classification model, the samples in each cluster generated by our unsupervised learning model are not clearly labelled. Therefore, the clustering model is not as straightforward and efficient as the classification model when user problems are assigned to different departments for processing.

# 5 Possible Future Work

## 5.1 Improvements of Language Classification

When we performed language detection and classification on the initial data, we found that a portion of the data was misclassified as other languages. As we model different-language data separately, language classification errors can directly affect the accuracy of our models and results. Based on this problem, improving the accuracy of language detection and classification is an important research and development direction for our ticket support system.

## 5.2 Improvement of Preprocessing

From the previous analysis, we found that the German model doesn’t perform as well as the English model. The main reason may be found in the preprocessing. Therefore, for German data we should adopt some more suitable preprocessing methods:

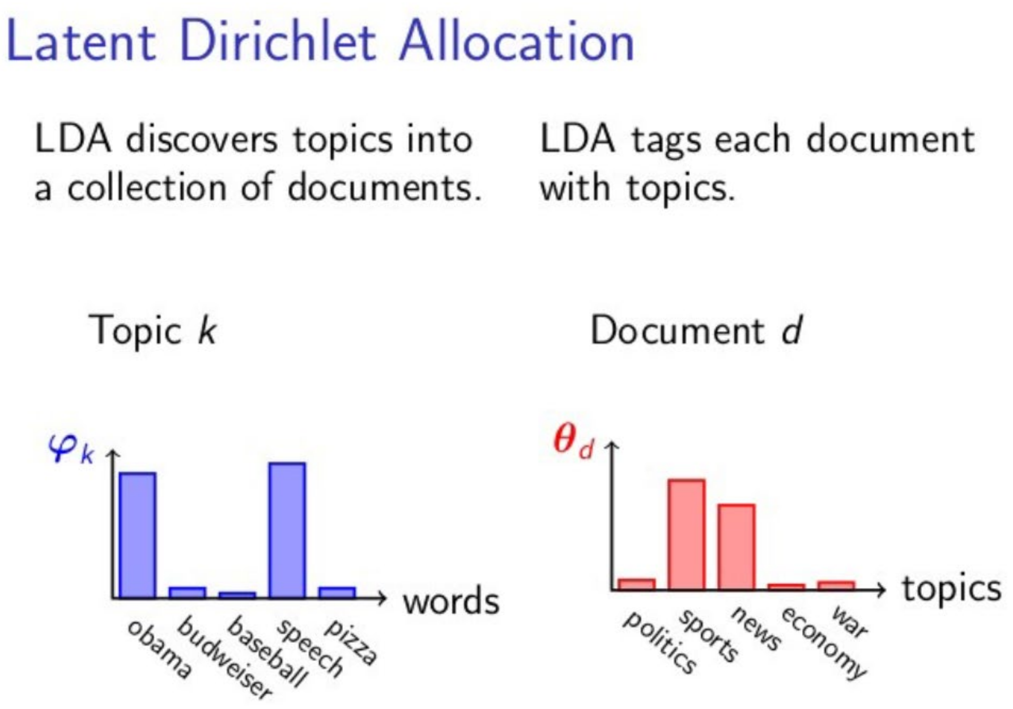
1) Use German-specific preprocessing libraries, such as "german_stopwords_full".

2) Use German-specific preprocessing methods, such as Part of speech tagging (POS), the Compound split of words, etc.

## 5.3 Improvement of the Model - LDA

The clustering model we used is the most commonly K-means algorithm. If we use another clustering model, we may achieve better clustering results. There are very promising developments in unsupervised learning, such as the application of neural networks to unsupervised learning, the use of LDA algorithm, etc. Here we perform a preliminary build of the LDA model.

Latent Dirichlet Allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. LDA is complicated in mathematical deriving, we just need to understand his features and some of the parameters associated with implementing them.


Latent Dirichlet Allocation

LDA discovers topics into a collection of documents.

LDA tags each document with topics.

Figure 5.1: Principle of Latent Dirichlet Allocation (LDA)

It is helpful to think of the entities represented by $\theta$ and $\varphi$ as matrices created by decomposing the original document-word matrix that represents the corpus of documents being modeled. In this view, $\theta$ consists of rows defined by documents and columns defined by topics, while $\varphi$ consists of rows defined by topics and columns defined by words. Thus, $\varphi_1, \dots, \varphi_K$ refers to a set of rows, or vectors, each of which is a distribution over words, and $\theta_1, \dots, \theta_M$ refers to a set of rows, each of which is a distribution over topics.

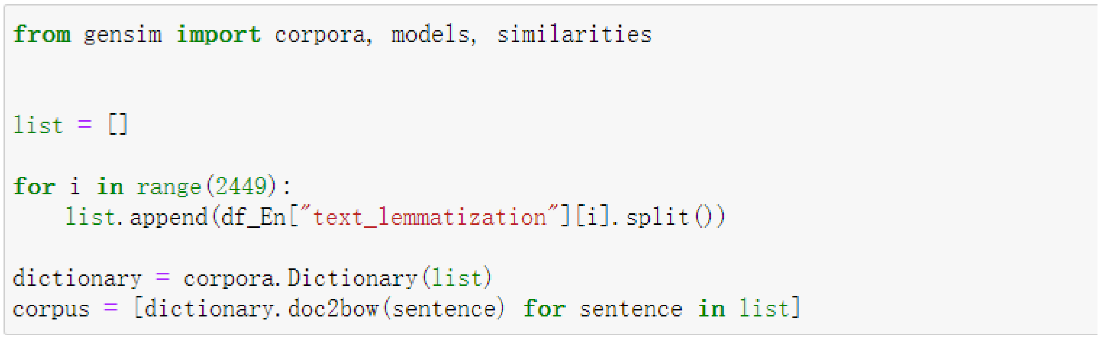

```
from gensim import corpora, models, similarities


list = []

for i in range(2449):
    list.append(df_En["text_lemmatization"][i].split())

dictionary = corpora.Dictionary(list)
corpus = [dictionary.doc2bow(sentence) for sentence in list]
```

Figure 5.2: Codes of LDA algorithm

We use the ready-made packages from genism to have a try on this text clustering task. By adding the lemmatized text to the list we get the input for LDA, namely the corresponding dictionary and corpus.

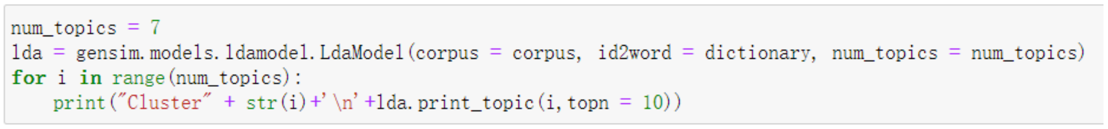
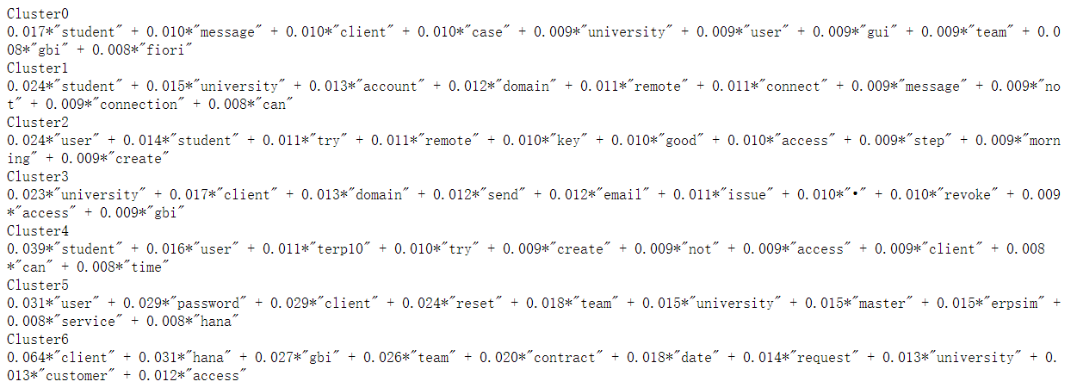

```
num_topics = 7
lda = gensim.models.ldamodel.LdaModel(corpus = corpus, id2word = dictionary, num_topics = num_topics)
for i in range(num_topics):
    print("Cluster" + str(i)+'\n'+lda.print_topic(i,topn = 10))
```

Figure 5.3: Codes of LDA algorithm

Similarly, we trained the LDA model also with 7 clusters according to the results in LSA and printed each cluster with the top 10 important words and their weights in the following figure:



```
Cluster0
0.017*"student" + 0.010*"message" + 0.010*"client" + 0.010*"case" + 0.009*"university" + 0.009*"user" + 0.009*"gui" + 0.009*"team" + 0.0
08*"gbi" + 0.008*"fiori"
Cluster1
0.024*"student" + 0.015*"university" + 0.013*"account" + 0.012*"domain" + 0.011*"remote" + 0.011*"connect" + 0.009*"message" + 0.009*"no
t" + 0.009*"connection" + 0.008*"can"
Cluster2
0.024*"user" + 0.014*"student" + 0.011*"try" + 0.011*"remote" + 0.010*"key" + 0.010*"good" + 0.010*"access" + 0.009*"step" + 0.009*"morn
ing" + 0.009*"create"
Cluster3
0.023*"university" + 0.017*"client" + 0.013*"domain" + 0.012*"send" + 0.012*"email" + 0.011*"issue" + 0.010*"•" + 0.010*"revoke" + 0.009
*"access" + 0.009*"gbi"
Cluster4
0.039*"student" + 0.016*"user" + 0.011*"terp10" + 0.010*"try" + 0.009*"create" + 0.009*"not" + 0.009*"access" + 0.009*"client" + 0.008
*"can" + 0.008*"time"
Cluster5
0.031*"user" + 0.029*"password" + 0.029*"client" + 0.024*"reset" + 0.018*"team" + 0.015*"university" + 0.015*"master" + 0.015*"erpsim" +
0.008*"service" + 0.008*"hana"
Cluster6
0.064*"client" + 0.031*"hana" + 0.027*"gbi" + 0.026*"team" + 0.020*"contract" + 0.018*"date" + 0.014*"request" + 0.013*"university" + 0.
013*"customer" + 0.012*"access"
```

Figure 5.4: Clusters after LDA algorithm

We found that at least for the English test our LDA model didn't perform well compared with the LSA model. One important explanation for that is LDA will perform better than LSA when we are accessing more ticket samples, for example, more than 100k. With more and more user data can be collected, our LDA model will own a better prospect in text clustering tasks.

# 6 Personal Evaluation

After working on this project for the last month, we are satisfied and happy with the structure of our project and the final model. Starting with the analysis of the requirements, we had a lot of brainstorming and discussions. Based on the suggestions of our supervisors, considering objective factors such as time, we worked together to develop the main structure and timeline of our project, and we basically completed our project as planned. Our main time is invested in the third stage, which is the modeling task part and also the main body of our project. A sensible division of labor kept our project moving smoothly. We have also systematically analyzed our model and the project as a whole from various perspectives and have finally identified possible future directions. Our truncated SVD model, as well as the clustering model, can be used as part of the preprocessing for subsequent work, such as a reference for the FAQs setting, providing a reference for the labels of the classification model, etc. Thanks to our supervisors, Simon Fuchs and Omar Shouman, for their help!