Master Thesis

# Automatic Topic Classification of Customer Feedback in the Banking Domain

Elena Theresa Weber

*a thesis submitted in partial fulfilment of the requirements for the degree of*

**MA Linguistics**
(Text Mining)

**Vrije Universiteit Amsterdam**

Computational Lexicology and Terminology Lab
Department of Language and Communication
Faculty of Humanities



Supervised by: Ilia Markov, Gabriele Catanese
$2^{nd}$ reader: Isa Maks

Submitted: July 1, 2022

# Abstract

This thesis project focuses on the Automatic Topic Classification of customer feedback derived from the banking domain. Topic Classification is the task of assigning a topic for a particular document from a set of predefined topics. For this, we explore and compare various conventional machine learning approaches (Support Vector Machine, Logistic Regression, and Naive Bayes) with more recent deep learning ones (BERT, RoBERTa, and DistilBERT) that currently provide the state-of-the-art results for a vast majority of Natural Language Processing tasks. Due to the unbalanced nature of the dataset, several topic adaptation, data augmentation and reduction methods are being tested. The data augmentation method focuses on back-translation. In contrast, the topic adaptation methods merge topics with overlapping content as well as topics that are inherently underrepresented within the dataset. Additionally, to determine the sufficient number of training examples needed for a classifier to provide a reasonable performance, we not only evaluate the performance of the merged datasets but also implement a data reduction approach in the form of undersampling.

The project showed that trained on the original dataset, Support Vector Machine with a Bag of Words TF-IDF feature representation provided the best performance with a macro-averaged F1-score of 0.537. Data augmentation improved the performance of BERT but not of RoBERTa or DistilBERT nor the traditional machine learning models. Additionally, it was determined that for this project concerning a transfer-learning approach compared to a traditional machine learning one, more training samples are needed. For transfer learning, a minimum of around 80 training examples is needed. In contrast, traditional machine learning models can identify topics after being trained on around 40, and if the train dataset is entirely balanced, around 20 samples are sufficient. Finally, this thesis project contributes to extending the research on Topic Classification of unbalanced datasets in the banking domain.

**Keywords** Text Mining, Natural Language Processing, Machine Learning, Automatic Topic Classification, Artificial Intelligence, Transformer, Deep-Learning, Bank Domain, CX, Unbalanced Data

# Declaration of Authorship

I, Elena Theresa Weber, declare that this thesis, titled *Automatic Topic Classification of Customer Feedback in the Banking Domain* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a degree degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: 29-06-2022

Signed: Elena Theresa Weber

# Acknowledgments

I want to thank my supervisors, Dr. Ilia Markov and Gabriele Catanese, for their unconditional support, guidance, and constructive criticism during this thesis project. Besides, I would also like to thank my amazing colleagues and dearest friends, Konstantina Andronikou and Rorick Terlou, for supporting me throughout this process. Additionally, I highly appreciate the help from the rest of my study group, Mira Reisinger, Mekselina Doganc, and Vicky Kyrmanidi, during the process of the thesis project but also throughout the whole master's program.
I am beyond grateful for the opportunity at Underlined and its staff for believing in me to execute this project. Furthermore, I would like to thank the CLTL staff of the VU Amsterdam for sharing their knowledge with me and supporting me during this master's.
Lastly, I want to thank my family and friends specifically. Without their unconditional support, love, and patience, I would not have been able to achieve this.

# List of Figures


1.1 Simplified Process of Automatic Topic Classification . . . . . 3

3.1 Distribution of the Topics . . . . . 14
3.2 Distribution of the topics after the dataset has been split . . . . . 16
3.3 Overview frequency after merging labels with overlapping content . . . . 18
3.4 Overview frequency after merging underrepresented topics into the topic "Other" . . . . . 19
3.5 Over- and undersampling (Al-Serw, 2021) . . . . . 19
3.6 Data augmentation using back-translation . . . . . 20
3.7 Naive Bayes formula (Loukas, 2022) . . . . . 25
3.8 S-curved graph of logistic regression, taken from (TowardAI, 2021) . . . 25
3.9 Hyperplanes using Support Vector Machine (Gandhi, 2018) . . . . . 26
3.10 Comparison Process Traditional Machine Learning and Transfer Learning 27
3.11 Transformer Architecture (Voita, 2022) . . . . . 28
3.12 Self-attention calculation in matrix form (Alammar, 2018b) . . . . . 29
3.13 Process multi-headed self-attention (Alammar, 2018b) . . . . . 30
3.14 Basic procedure BERT, adapted from Alammar (2018a) . . . . . 31

4.1 Confusion Matrix SVM BoW TF-IDF Original Dataset . . . . . 41
4.2 Confusion Matrix SVM BoW TF-IDF Back-translation 10% Dataset . . 47
4.3 Confusion Matrix SVM BoW TF-IDF Back-translation 20% . . . . . 48
4.4 Confusion Matrix BERT Back-translation 20% . . . . . 49
4.5 Confusion Matrix RoBERTa Merged Dataset . . . . . 53
4.6 Confusion Matrix RoBERTa Other Dataset . . . . . 56
4.7 Individual F1-scores of the topics in relation to their sample size . . . . 59
4.8 Confusion Matrix Logistic Regression BoW TF-IDF undersampled train set . . . . . 59

# Contents

**Abstract** i

**Declaration of Authorship** iii

**Acknowledgments** v

**List of Figures** vii

**1 Introduction** 1
- 1.1 Problem Definition . . . 2
- 1.2 Research Question(s) . . . 3
- 1.3 Approach . . . 3
- 1.4 Outline of the Thesis . . . 4

**2 Related Work** 5
- 2.1 Task Description: Automatic Topic Classification . . . 5
- 2.2 Approaches . . . 5
  - 2.2.1 Rule-based . . . 6
  - 2.2.2 Machine Learning . . . 6
  - 2.2.3 Transfer Learning . . . 8
- 2.3 Unbalanced data distribution . . . 9
- 2.4 Bank Domain . . . 10
- 2.5 Concluding Remarks . . . 10

**3 Methodology** 13
- 3.1 Data . . . 13
  - 3.1.1 Stratified Data Splitting . . . 16
  - 3.1.2 Data Adaptation . . . 17
- 3.2 Automatic Topic Classification . . . 21
  - 3.2.1 Machine Learning . . . 21
  - 3.2.2 Transfer Learning and Fine-Tuning . . . 27

**4 Results** 35
- 4.1 Results . . . 36
  - 4.1.1 Results: Original Dataset . . . 36
  - 4.1.2 Results: Data Augmentation . . . 37
  - 4.1.3 Results: Merging Topics . . . 38
  - 4.1.4 Results: Data Reduction . . . 40
  - 4.1.5 Results: Concluding Remarks . . . 40

4.2 Evaluation of the Results - Error Analysis . . . 41
4.2.1 Analysis: Topic Classification on the Original Dataset . . . 41
4.2.2 Analysis: Topic Classification on Augmented Datasets . . . 46
4.2.3 Analysis: Merging Topics . . . 52
4.2.4 Analysis: Data Reduction . . . 58
4.2.5 Analysis: Overall Concluding Remarks . . . 61

**5 Conclusion and Discussion** **63**
5.1 Concluding Remarks about the Research . . . 63
5.1.1 Research Questions . . . 63
5.2 Future Work . . . 64
5.3 Limitations . . . 66

**A Results Traditional Machine Learning** **67**
A.1 Bag of Words and TF-IDF . . . 67
A.1.1 Original Dataset . . . 68
A.1.2 Merged Dataset . . . 69
A.1.3 Merged "Other" Dataset . . . 70
A.1.4 Back-translation 10% . . . 71
A.1.5 Back-translation 20% . . . 72
A.1.6 Undersampled Dataset . . . 73
A.2 Embeddings . . . 74
A.2.1 Original Dataset . . . 74
A.2.2 Merged Dataset . . . 75
A.2.3 Merged "Other" Dataset . . . 76
A.2.4 Back-translation 10% . . . 77
A.2.5 Back-translation 20% . . . 78
A.2.6 Undersampling . . . 79

**B Results Transfer Learning** **81**
B.1 Original Dataset . . . 82
B.2 Merged Dataset . . . 83
B.3 Merged "Other" Dataset . . . 84
B.4 Back-translation 10% . . . 85
B.5 Back-translation 20% . . . 86
B.6 Undersampled Dataset . . . 87

# Chapter 1

# Introduction

Envision working for a company that provides different products and services. After customers make a successful transaction, the company wants to evaluate its performance, services, and the overall experience of the customer. Customer Experience (CX) is a crucial factor for a successful business as it focuses on the cognitive, emotional, behavioral, sensorial, and social responses of a customer concerning the services of a company (Lemon and Verhoef, 2016). The company wants to work with the CX and starts sending out surveys via email to detect aspects needing improvement. The form begins with rating scales to extract a satisfaction score. This score provides a valuable first insight into the CX with the company and its services. It is a simple way to receive an initial overview but does not give the customer the option to share their personal experience and sentiment toward the company in detail. For a better understanding of the CX, user-generated content (UGC) is being requested in the survey to retrieve this information. This kind of feedback comes in text form and enables customers to provide a more personalized opinion and include aspects that have not been asked in the survey so far. Apart from filling out surveys, customers have additional touch-points with companies and can share their opinion about their experiences easily online. This offers companies a new way of gathering feedback on their services and products and can thus be used to improve the customer's experience for future services. After some time, enough data for an evaluation is retrieved through the survey or the user-generated online feedback. However, after that, what can be done with it? One could read the feedback one after another, analyze it, and take notes about the insights provided through the UGC. By analyzing and using text data, a lot of more profound value is being created for companies. Doing this manually is not only immensely time-consuming and a waste of resources but only possible at a large scale by applying automated methods. Luckily, different approaches and methods exist to access the value hidden within text data.

To extract this sort of value, Natural Language Processing (NLP) can be of assistance. NLP combines linguistics with computer science and artificial intelligence and teaches models and computers to understand and work with language data. It relies on "formal models, or representations, of knowledge of language at the levels of phonology and phonetics, morphology, syntax, semantics, pragmatics and discourse. A number of formal models including state machines, formal rule systems, logic, and probabilistic models are used to capture this knowledge." (Jurafsky and Martin, 2009, p.15). The goal of the thesis project is the implementation of an NLP task to CX. More specifi-

cally, to evaluate the performance of state-of-the-art models with regard to the specific features of the dataset provided for this project.

## 1.1 Problem Definition

The thesis project aims to implement an NLP task to customer experience. The data for this project was collected via email surveys that customers of a major Dutch bank have filled out. The experiments are carried out in collaboration with Underlined[1], a Dutch software company providing solutions for CX Analytics and linking customer insights with company performance. For their tools, since CX also comes in the form of text data, the implementation of NLP is a necessity.
Since the data is derived from a specific domain and thus contains very similar content in each feedback, one of the first steps is to group the data into different categories. Some feedback might be about the *Digital Options* or the *Employee Knowledge & Skills* whereas other customers just mention their *Overall Experience*. By doing so, companies have an insight into the performance of their different services - the frequency of customer feedback concerning one service could, for instance, indicate a need for improvement. This can, of course, be done manually by reading the feedback sentence by sentence and then classifying it into a specific group - a rather time-consuming and inefficient way to deal with it. Nowadays, and especially due to the rapid technological development, companies can receive feedback in high amounts daily. This makes it impossible to classify the feedback manually, and besides being extremely time-consuming, it is economically inefficient. However, this process can be automatized using NLP tasks like Automatic Topic Classification (TC). Topic Classification is considered one of the oldest NLP tasks with the goal of assigning a label or a topic to a text or document (Jurafsky and Martin, 2009). Henceforth, the project aims to build and improve Underlined's tool by providing an automated classification of feedback into preset categories, also called topics.
Since the project handles feedback and highly domain-specific data, the feedback comes with additional challenges. Real-life data is often not only noisy but also highly unbalanced. Unbalanced here means that the frequency of the various topics within the dataset is not equally distributed, and some topics appear way more often compared to others. That can lead to confusion for the classification algorithms and thus wrong predictions. Additionally, the unavailability of high-quality annotated data contributes to those challenges. Customer feedback differs from other data in their text form structurally and stylistically. For instance, some customers provide long and extensive feedback while others provide one or two words. The data handled in this project was translated from Dutch into English and consists of asynchronous sequences since no interaction took place to retrieve the feedback. It is unbalanced in the sense of having a high number of various topics and the representation varying from a few to several thousand. This unbalancedness arises another challenge for Topic Classification as well as other NLP tasks, as it can confuse the models and affect their performance. This needs to be considered when working with UGC and CX. One sub-task of this project consequently focuses on the unbalancedness by experimenting with augmenting the labels in a way that does not affect nor change the original data and its labeling. Due to its domain-specificity with a focus on the banking domain, results from related works are only partially eligible for comparison. Another challenge this project faces is the

[1] https://underlined.eu/

limited time and resource availability. Our results provide valuable insights on TC and highlight important directions for future work.

## 1.2 Research Question(s)

The extensive examination of the data and the related works led to the following research questions:

**Research Question:** Which classifier provides the best results for the Automatic Topic Classification task on customer feedback in the banking domain?

**Subquestion:** Will data augmentation techniques improve the classifiers' performance in the banking domain?

**Subquestion:** What is the minimum required number of training examples within each category for a classifier to provide a reasonable performance?

## 1.3 Approach

The project focuses on comparing the performance of different classifiers using all categories provided in the dataset and several experimental set-ups, changing the topics by merging and using data augmentation techniques. For the classification task, we will evaluate and compare conventional machine learning approaches, Support Vector Machine (Cortes and Vapnik, 1995), Naive Bayes (Bayes, 1763), and Logistic Regression (Cox, 1958), with transformer-based pre-trained language models, BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and DistilBERT (Sanh et al., 2019), that are fine-tuned on the data distributed. Figure 1.1 provides a simplified overview of the proceedings of the project.

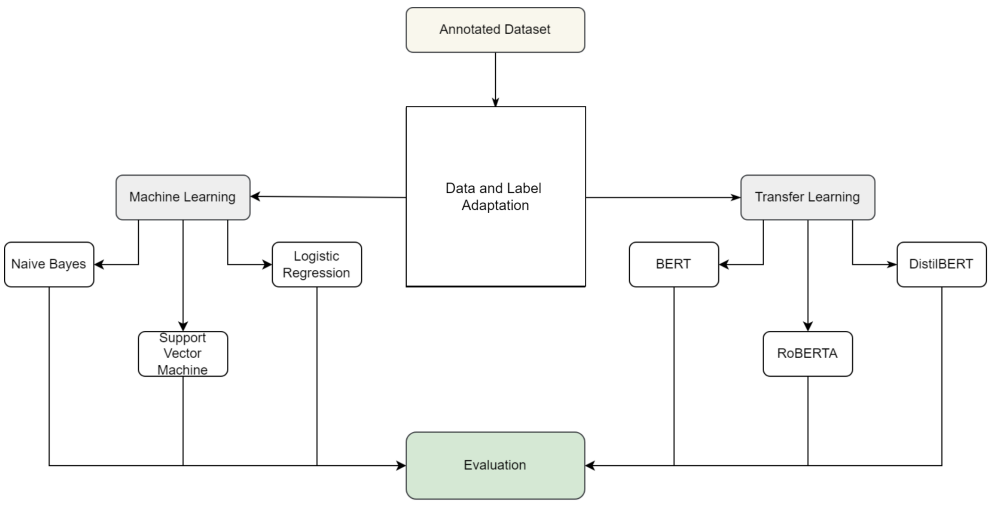



Figure 1.1: Simplified Process of Automatic Topic Classification

The conventional machine learning approaches will additionally be explored using several pre-processing steps. The project aims to evaluate the performance of state-of-the-art NLP models in the banking domain with regard to limited training data,

unbalanced distribution of a large number of classes, as well as the unequal length of feedback statements, and, ultimately, select the model providing the best evaluation results. In the next step, the minimum required examples of labeled classes are evaluated to receive an adequate classification performance are evaluated. This can be useful for future projects to comprehend how many examples per labeled class are needed for a classifier to provide reliable results in the banking domain using a skewed dataset.

## 1.4 Outline of the Thesis

The chapters of this thesis are segmented as follows. *Chapter 2* provides a general outlook in recent literature about Customer Experience and TC, focusing on the different Machine Learning and Transformer-based approaches. *Chapter 3* gives an introduction to the data and presents the methods behind the experiments. The chapters mentioned so far are followed by *Chapter 4*, presenting the results of the experiments which are being discussed and evaluated in an error analysis. Lastly, *Chapter 5* provides an overall conclusion and insight into this project's limitations and future directions.

# Chapter 2

# Related Work

The following sections provide an overview of the most recent related works in TC focusing on customer feedback. At first, describing the task of TC and then the various approaches based on traditional machine learning and transfer learning.

## 2.1 Task Description: Automatic Topic Classification

TC is also often referred to as Topic Detection and Text Categorization. As the name of the task already implies, it is a classification task. The task focuses on assigning labels or categories to a sentence-, text-, or document-level (Jurafsky and Martin, 2009). Since the focus of this project is on UGC distributed as feedback statement, the goal is to identify a topic of a feedback with a predefined topic (Menner et al., 2016).
For this task, supervised machine learning techniques are applied. Already labeled data is used to train a classifier to predict said labels based on observations within the data. The model takes input x and a fixed set of output classes $Y = y1, y2, ...yM$ from the training data and then predicts classes y $\epsilon$ Y on the test data. To put it more into perspective, the models need to be able to predict a predefined topic after feeding a feedback sequence into it. Given the sentence "The app works well and I can do all my transactions with it!" the model has to be able to detect the predefined topic, in this case, *Digital Options*. Depending on the data and the task, classification tasks can come in various shapes. In the case of only having two classes, the task is called binary classification. Since this project focuses on data with several labels, it is considered a multi-class classification task on user-generated feedback sequences. TC can be used as a task on its own as well as combined with a variety of other NLP tasks, such as Sentiment Analysis, Opinion Mining, Email Classification, Spam Filtering, and Document Organization (Aggarwal and Zhai, 2012). In the case of this project, the focus is solely on TC - comparing several traditional machine learning algorithms with transfer learning algorithms. Possible approaches are explained in further detail in the sections below.

## 2.2 Approaches

In order to perform a TC, various techniques have been developed. One can write a classification model from scratch using rule-based methods, traditional machine learning, or transfer learning. This section first introduces rule-based approaches, then focuses on machine learning followed by transformer-based methods.

### 2.2.1 Rule-based

By using a rule-based approach, a classification model is entirely built from scratch without making use of pre-existing models or techniques. Instead, one uses hand-made rules created after extensively going through the data and its annotation and thus gets a deeper understanding of it. To be more precise, one set of rules corresponds to a word pattern and the other one to the labels of the topics (Aggarwal and Zhai, 2012). One approach for ATC, or in their case, frequent feature identification, is implementing association mining Hu and Liu (2004). This is used to trace the frequent set of words or phrases occurring within a sentence since those words or phrases are most likely related to specific topics. Apart from using a rule-based approach on its own, it can also be combined as a feature within machine learning models (Brun et al., 2014; Saias, 2015), as a hybrid (Carenini et al., 2005), or deep learning approaches on opinion mining (Ray and Chakrabarti, 2020).
Rule-based models can be improved over time by updating existing rules as well as implementing new rules. However, this approach requires deep knowledge and understanding of the data and the domain. Overall, they are very time-consuming in their creation. Additionally, it is hard to keep them updated, and implementing new rules might downgrade the performance in the long run.

### 2.2.2 Machine Learning

In short, a machine learning classification model learns patterns from a set of annotated training data. The text data has to be transformed into vectors for the algorithm to work with it. The algorithm then makes predictions about unlabeled data by generalizing those learned patterns. To create the classifier model, one can make use of different machine learning algorithms that each have their advantages and disadvantages. To create the vector representation, one can, for instance, make use of Bag of Words or an embedding representations. This section presents recent research on TC using the most common machine learning approaches.

A Naive Bayes (NB) classifier is a probabilistic learning method that is based on the Bayes theorem (Spasić et al., 2012). It is called naïve because it makes a simplifying assumption about the interaction of features Jurafsky and Manning (2012). The model assumes that features are independent of each other. Even though NB is often seen as a low-performance classifier, it has been proven effective, especially for text classification (Rish et al., 2001). For TC on suicide notes, Spasić et al. (2012) used a multinomial Naive Bayes classifier. In this approach, a sentence is classified with a single topic using the topic that is suggested by the most considerable posterior probability when its value exceeds a specific threshold (Spasić et al., 2012). Spasić et al. (2012) additionally proved that NB does not necessarily require a large amount of training data for a good performance. A more related task was approached by Menner et al. (2016), identifying relevant topics in tourism reviews. In their research, the best performing model on topic classification was a Naive Bayes in combination with a Named Entity Recognition, having a precision of 73.84% for topic detection (Menner et al., 2016).

Support Vector Machines (SVM) (Cortes and Vapnik, 1995) were introduced as a kernel-based machine learning model for classification and regression tasks. Because they are so extraordinarily generalizable (Cervantes et al., 2020), they are often treated

as the most powerful and commonly used classifiers. An SVM aims to find an optimal separating hyperplane that maximizes the margin of training data (Jurafsky and Manning, 2012). This works by partitioning the data space using linear or non-linear delineations between the classes and then calculating the optimal boundaries between the classes (Aggarwal and Zhai, 2012). In a classification task, "when a new data point is mapped into the original vector space, the average distances between the new data point and the support vectors from different categories are measured using the Euclidean distance. The classification decision is made based on the category of support vectors with the lowest average distance with the new data point, making the classification decision irrespective of the efficacy of hyper-plane formed by applying the particular kernel function and soft margin parameter." (Cervantes et al., 2020, p.200). Comparing Support Vector Machines with Naive Bayes, it can often be noted that SVM outperforms the NB. However, it has to be mentioned that this is always affected by the amount of training data and the task itself. In a research article about classifying customer opinions from Twitter as positive and negative, SVM scored an accuracy of 83.34% whereas the NB scored 75% (Kusumawati et al., 2019). By classifying reviews in the restaurant domain with an SVM, Kiritchenko et al. (2014) obtained the first rank among 21 other submissions with an F1-score of 88.85%. They optimized the parameter C with cross-validation separately for each classifier in their approach. Additionally, the sentences were tokenized, stemmed, and several n-gram-related features were implemented (Kiritchenko et al., 2014). In general, it is said that an SVM does not perform well on large data sets, unbalanced data sets, and multi-classification. They were initially built to solve binary classification problems and have their limitations in parameter selection and algorithmic complexity (Cervantes et al., 2020).

Logistic Regression, sometimes called Maximum Entropy classification, is a discriminative classifier that learns which features are the most useful to discriminate between the possible classes. It is also often described as the baseline in supervised machine learning algorithms for classification tasks (Jurafsky and Manning, 2012). Generally speaking, it is more often used when the target variable being learned is numerical as opposed to categorical (Aggarwal and Zhai, 2012). On an approach of classifying tweets into their four most commonly discussed topics (Indra et al., 2016), the researchers scored an accuracy of 92% on a dataset containing 1800 tweets. Furthermore, there even is a software called SUR-Miner (Gu and Kim, 2015) which provides a summary of a users sentiment, opinion, and emotion on different topics.

Apart from the prominent Support Vector Machine, Logistic Regression, and Naive Bayes, other models can also be implemented. For instance, besides Naive Bayes and Support Vector Machine, Menner et al. (2016) additionally implemented k-nearest neighbor (Fix and Hodges, 1989) and Conditional Random Fields (Lafferty et al., 2001). Toh and Su (2015) use a sigmoidal feed-forward network to train binary classifiers for an aspect category classification and a Conditional Random Field in addition to that for opinion target extraction. Besides, deep learning, inspired by the workings of human neurons, is also often used in TC in the form of neural networks (Aggarwal and Zhai, 2012). Stanik et al. (2019) implemented a deep convolutional neural network (CNN) to classify English and Italian feedback about apps, whereas Aslam et al. (2020) created a model that classifies app reviews using non-textual information and a CNN.

In order to vectorize the text and thus make it understandable for the classification models, Bag of Words (BoW), TF-IDF, or embeddings can, for instance, be implemented. A BoW creates a dictionary with all the terms that appear within the feedback sequences and, as a next step, calculates how often represented the term is in a feedback and how it is labeled. The weight of a word is determined by its frequency. Maalej et al. (2016) showed the effectiveness of using a Bag of Words approach on the binary classification of app reviews using Naive Bayes with a precision of 70% on the pure text classification. Once the approach has been fine-tuned, the precision increased up to 92%.
Alternatively, one can use term frequency-inverse document frequency, in short TF-IDF, by combining the frequency of the terms with the inverse document frequency to get a score for the importance of a word within a corpus. Na et al. (2004) compared TF-IDF with term presence and term frequency and combined it with Support Vector Machine for an Automatic Sentiment Classification of product reviews. The researchers proved the effectiveness of TF-IDF. It can additionally be used as an extension of BoW Rustam et al. (2021).
Since Bag of Words and TF-IDF do not capture semantic information nor the distance between words, embeddings in the form of mapped vectors in a low dimensional space can be used to comply with this informational issue (Mottaghinia et al., 2021). By using traditional embedding models such as Word2Vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014) the classifier has a better understanding of the overall meaning of the text and thus improves the performance. Embedding models have been used in previous research about customer-experience (Borg et al., 2021) showing that an LSTM combined with GloVe outperforms other non-sequential models when predicting labels for e-mails.

### 2.2.3 Transfer Learning

Implementing deep learning methods has the advantages that the models alleviate the feature engineering problem, do not rely heavily on handcrafted features, and make use of low-dimensional and dense vectors (Qiu et al., 2020). However, their large number of needed parameters, the chance of overfitting on small training data, and having trouble with generalization create an issue when implementing them on NLP tasks.

To tackle these problems, pre-trained models can be of help: the first generation consists of pre-trained static and contextual word embeddings (Hadi and Fard, 2021), such as Glove (Pennington et al., 2014) and word2vec(Mikolov et al., 2013). They are quite effective but context-independent, in most cases trained on shallow models and learn everything from scratch (Qiu et al., 2020). Oftentimes they also and fail to model polysemous words (Hadi and Fard, 2021). The second generation however uses pre-trained contextual encoders (Qiu et al., 2020), where neural encoders are pre-trained on a sentence-level or even higher and thus learn contextual embeddings in the form of context-sensitive word representations (Hadi and Fard, 2021). By training those models on large corpora, they are able to learn universal language representations, i.e., the implicit linguistic rules and common sense that is hidden in text data in the form of lexical meanings, syntactic structures, semantic roles, and pragmatics with the core idea of describing the meaning of a text sequence in the form of low dimensional vectors (Qiu et al., 2020). This is also called transfer learning. In recent years, the development of transformer-based language models achieved immense success with their

enhanced architecture and are able "to learn universal language representations from large volumes of unlabeled text data and then transfer this knowledge to downstream tasks." (Kalyan et al., 2021, p.2).
Transformers (Vaswani et al., 2017) were first introduced in 2017 as a deep-learning model based on self-attention that contains a stack of encoder and decoder layers. With those layers, transformers learn the language information and its complexity (Kalyan et al., 2021). Pre-trained transformer models (PTM) (Hadi and Fard, 2021) have been trained on large corpora and are able to support various NLP tasks such as TC through fine-tuning. Their architecture is being explored in greater detail in Chapter 3. Using an existing PTM is less time-consuming as no new models have to be created from scratch. The models transfer the learned knowledge from a corpus to a new domain, or task (Hadi and Fard, 2021).
Transformers can be implemented as a classification tool, and Hadi and Fard (2021) confirmed that PTMs achieve higher scores in classification tasks compared to prior approaches. For TC, it is recommended to use auto-encoding transformer models, such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and DistilBERT (Sanh et al., 2019). A recent project about customer reviews in the electric vehicle domain (Ha et al., 2021) proves the effectiveness of using BERT within TC, scoring an accuracy of 91%. In other projects, a BERT-based sequence classifier achieved a classification accuracy of 87% concerning feedback analysis (Mekala et al., 2021). Additionally, transfer learning and TC is used in combination with other NLP tasks, such as aspect-based sentiment analysis (Sun et al., 2019).

## 2.3 Unbalanced data distribution

Working with natural data always comes with the issue of the datasets being unbalanced, skewed, and noisy, thus affecting the performance of the classifiers. The following approaches are not directly related to Automated Topic Classification but can be implemented in order to improve the performance of a classifier working with an unbalanced data distribution.

Different approaches have been proposed to handle the data disparity. There are two main approaches to resolve the problem, modifying the classifier or modifying the data (Mountassir et al., 2012). To modify the classifier cost-sensitive learning (Brank et al., 2003) as well as a class-boundary-alignment algorithm (Wu and Chang, 2003) can be implemented. The main takes on modifying the data are undersampling (Kubat et al., 1997) and oversampling (Chawla et al., 2002). The goal of undersampling is to decrease the sentences that are part of an overly represented topic, whereas oversampling tries to increase the sentences within underrepresented topics. A common approach for undersampling is random undersampling (Li et al., 2011; Burns et al., 2011). Mountassir et al. propose three different undersampling techniques. First, *remove similar*, i.e., removing documents from an overly represented topic that are very similar to other documents in said topic. This way, the dataset becomes more balanced but does not lose much information. Second, *remove farthest*; here, the documents within an overly represented topic are eliminated when they differ greatly from the others in their content. Lastly, *remove by clustering* works by implementing a clustering algorithm on the overly represented topics to create an optimally balanced dataset (Mountassir et al., 2012). By removing unbalancedness and sparsity to something more manageable and

meaningful (Damaschk et al., 2019), it was shown that the classifier's performance can be improved substantially. Apart from over- and undersampling, other approaches to modify the data could be data augmentation. Data augmentation is mostly used for image data, for instance, by slightly rotating the image or changing the color distribution. Applying augmentation techniques to text data is rather difficult, as it affects the semantics and thus the meaning of the text data. However, it has been proven to enhance the performance of NLP tasks. Some are back-translation, lexical substitution, transforming the text surface, injecting random noise, syntax-tree manipulation, and mixing up texts (Chaudhary, 2020).
Additionally, it has been proven that another factor for a poor performance is not necessarily the unbalanced distribution of the data but the overlapping of the classes (Prati et al., 2004) - confusing the models and their predictions. Another approach, that is also done to unbalanced data and is especially done using image data, is merging near-identical classes into a single merged class in order to reduce the overall number of classes and unbalancedness of the class distribution (Potyraj, 2021).

## 2.4 Bank Domain

A selected number of papers focused explicitly on classification tasks in the banking domain, and to our knowledge, barely any paper focuses specifically on TC concerning user-generated content derived from surveys - making it an underrepresented task within the domain. Instead, many make use of online resources, like Twitter, to derive user-generated content about banks and classify them based on the corporate reputation (Rantanen et al., 2019), their service attributes in combination with a sentiment analysis (Mittal and Agrawal, 2022), or opinion mining with sentiment analysis (Chaturvedi and Chopra, 2014). Several approaches focus solely on sentiment analysis instead of topic classification (Kazmaier and Van Vuuren, 2020; Krishna et al., 2019; Leem and Eum, 2021; Raicu, 2019), comparing traditional machine learning algorithms with neural network architectures. Other approaches focus on data derived from internet forums of a specific bank and use TC on top of a topic modeling task (Vencovskỳ et al., 2016).
Apart from analyzing customer feedback, NLP tasks are applied in fields like financial predictions, for instance, predicting the stock market, banking, in the form of money laundering detection, and corporate finance as a means of analyzing reports or detecting fraud (Gupta et al., 2020).
Furthermore, it has to be mentioned that most researchers behind the papers do not come from a linguistic background but rather an engineering (Kazmaier and Van Vuuren, 2020), banking (Krishna et al., 2019), economics (Rantanen et al., 2019; Raicu, 2019; Vencovskỳ et al., 2016), and marketing background (Mittal and Agrawal, 2022) and thus oftentimes lack a deeper understanding of NLP and its techniques.

## 2.5 Concluding Remarks

The most common and successful approaches concerning classification models as well as their effectiveness were presented in the sections above. For this project, it has been decided to compare conventional machine learning approaches with different transformer-based models.
Concerning machine learning, we will be using Support Vector Machine, Naive Bayes, and Logistic Regression. The reasons why the other classifiers were not considered are

the following. K-nearest neighbors are easy to implement but are slow when it comes to extensive input data. Additionally, they are quite sensitive regarding irrelevant parameters (Cervantes et al., 2020). Concerning decision trees, which are the hierarchical decomposition of the data space (Aggarwal and Zhai, 2012), they are usually fast in the training phase but are not flexible when it comes to modeling parameters (Cervantes et al., 2020). For the transformer-based models, it has been decided to compare the performance among BERT, RoBERTa, and DistilBERT. The machine learning models and the architecture behind the transformer-based models are explained in greater detail in Chapter 3. All experiments will be conducted under the consideration of the unbalanced data distribution by including data and topic adaptation methods and keeping in mind the specificity of the vocabulary used within the data derived from the banking domain.

# Chapter 3

# Methodology

This chapter presents the methodology behind the project. Starting with a thorough description of the data and the content of each topic, how the data is being adapted to create the experimental setups, followed by a description of the machine learning and the transfer learning approaches[1].

## 3.1 Data

The data that is used for this project is provided by de Volksbank[2] and consists of customer feedback in text form in English (**ISO 639-1 en**). The intended audience is thus the company that receives the feedback with the goal of optimizing its services. The feedback was generated by sending out surveys via email to customers affiliated with de Volksbank concerning mortgages. They included open questions as well as rating scales.
Once the data was retrieved, it was machine translated[3] from Dutch to English as the system that is written for Underlined is at first targeted to work in English before other languages. The data was then manually checked by two working students from Underlined who are both native Dutch speakers and have a bilingual proficiency in English. They are familiar with NLP and the task of TC. The data consists of 6,071 rows, each containing customer feedback with a mean length of 12.46 tokens per sentence. The maximum number of tokens within one feedback is 64, and the minimum number of tokens is 1, containing feedback like "okay", "fine", and "thanks". As it can be seen, some feedback statements are concise and cannot detect a topic, for instance "nothing to complain". This can be interesting for a sentiment analysis in future steps. Other feedback sequences are lengthy and elaborate, mentioning several topics within their review. The data was systematically annotated with Dutch labels in a rule-based approach under human supervision that were later also machine translated into English labels. In total there are 11 main topics, the distribution can be seen in Figure 3.1.

[1] The repository for the project can be found here: `https://github.com/cltl-students/Weber_Elena_Automatic_Topic_Classification`

[2] `https://www.devolksbank.nl/`

[3] `https://cloud.google.com/translate`


| English labels | Total |
|---|---|
| Information Provision | 2012 |
| Employee Attitude and Behavior | 1186 |
| Handling | 664 |
| Overall Experience | 612 |
| Employee Knowledge & Skills | 481 |
| Processes | 477 |
| No topic found | 277 |
| Digital Options | 159 |
| Price and Quality | 115 |
| Employee Contact | 57 |
| Quality and Offering | 31 |
| Total number of labels | 11 |

Figure 3.1: Distribution of the Topics

In order to mitigate system bias and enable better science while alleviating issues related to exclusion and bias in NLP, Bender and Friedmann introduced Data Statements Bender and Friedman (2018). The following sections try to answer all issues mentioned by them.
It can be assumed that for most participants the native language is Dutch. Additionally, it has to be mentioned that due to the machine translation into American English, a lot of Dutch variety, like particular dialects, Dutch abbreviations, and colloquial variation, as well as the usage of idioms, gets lost. Because of that, the voice of the feedback sometimes shifts slightly. Spelling and grammar mistakes are primarily lost in the translation process. Some feedback data contains emojis, which are kept in the English translation. Characteristically, the vocabulary is frequently domain-specific, using words related to the banking domain and the offers they have. The structure of the data is also distinctive, as it seems like the customers were asked a specific question to share their experience with de Volksbank.
There is no information about the demographic of the speakers, respectively writers, of the feedback as it is anonymous. Since the feedback statements focus on the banking domain, it can be assumed that the participants are of legal age. However, the data does not include information about the overall age distribution. Equally, there is no information about the gender nor the socioeconomic status of the participants, meaning the distribution is most likely unbalanced and not representative. Some data sequences contain private information, like names and email addresses.
The data shall not be shared or distributed elsewhere and can only be used within the company.

### Topic Content

For a deeper understanding of the data and its content, the following paragraphs summarize the content of the eleven topics. This is especially needed in order to decide which topics have overlapping content and thus can be merged. The topics are presented in the order they appear in concerning their frequency.

The most represented topic within the dataset is *Information Provision*. As the name indicates, the topic focuses on the provision of information within conversations with mortgage advisors from de Volksbank and the information provided in additional

materials like documents. A typical feedback sequence for this topic could be "Clear conversation with the advisor, provided us with all the information we needed."[4] As it can be seen in the example, the feedback describes the experience with the advisor and also mentions the keywords *information* and *provided*.

*Information Provision* as the majority topic is followed by the topic *Employee Attitude and Behavior*. This topic describes the behavior of the employees toward the customers in different settings. An example of a feedback statement could be "Our bank has a very friendly personnel and they have always time for a pleasant and helpful conversation.", indicating that the personnel, in other words, the employees, behave in a friendly, pleasant, and helpful manner during an encounter.

The topic *Handling* describes how the bank and its' employees handle situations, requests, and conversations. "Everything is handled smoothly and clearly arranged." - describing that de Volksbank and its personnel are able to handle situations smoothly and arrangements clearly.

The *Overall Experience* summarizes the customers' general experience during their encounter with de Volksbank. This topic is beneficial to label feedback sequences that talk about the broader opinion of customers, describing that "Everything went well." or "I have always been satisfied with de Volksbank."

*Employee Knowledge Skills* focuses specifically on describing the customer's perspective on what the employees seem to know and how skillful they are when providing a service. In a feedback sequence, this could be phrased like this: "Our advisor is always prepared and helps solving our problems. Great advice!"

Many customers also mention the process of their requests and the service. This can contain the speed but also the way processes work overall. For this category, the topic *Processes* classifies fitting feedback sequences, like "Everything runs smoothly and we have been very satisfied with the process." into its topic.

*No topic found* is quite self-explanatory as a topic. It contains those feedback sequences that do not fit into one of the provided topics but also those that do not contain helpful or interpretable feedback. These could be feedback sequences that only contain one word, like "okay", "thanks", or "nice" - or longer sequences like "The meeting has not taken place yet - why did I even get this survey?". This topic is expected to lead to a lot of confusion for the models as it does not have one coherent content.

*Digital Options* is used for feedback sequences that specifically mention digital services offered by de Volksbank. This could be, among others, the website, online banking, an app, or email contact. Typical feedback might be "I can take care of everything online. I really like the App but the website could be improved."

Since this project is about the bank domain, some topics are also specifically about the *Price and Quality* of the offers. This topic contains customer experience about the

[4]The examples were manually created and not derived from the data. They have been created inspired by the original data as the data cannot be published.

calculated costs, the interest rate, the rate reduction, prices, and mortgages. The experience with the price and quality of their services could be described as "The interest that I have to pay for my mortgage is way too high. I am not happy with it."

The next-to-last topic concerning their number of representations within the dataset is *Employee Contact*. Customers describe how they experienced the contact and the contact points with employees. This could be in person, via email, in a phone call, or a video call. Often it is mentioned that employees respond quickly and provide a clear response, "They replied quickly. The employees are easily accessible and always offer a clear response."

Lastly, *Quality and Offering* focuses on the offers and their quality proposed by de Volksbank's representatives concerning, for instance, mortgages and repayments. A typical feedback sequence within this topic could be: "I can repay my mortgage monthly but I do not understand the additional fine."

### 3.1.1 Stratified Data Splitting

For this project, the data has been split using the ratio 80-10-10. A stratified splitting was used in order to preserve the proportion of examples within each topic in the different datasets. 80% of the whole dataset is now part of the train set, and the remaining 20% are split into test and validation sets. Meaning of the overall dataset 10% is now the test and the other 10% part of the validation set. The validation set is created by halving the test set and thus resulting in a validation and test set that both contain 10% of the data. The splitting has been achieved using scikit-learn (Pedregosa et al., 2011) and an overview of the distribution after the splitting can be seen in Figure 3.2 as well as in Table 3.1.

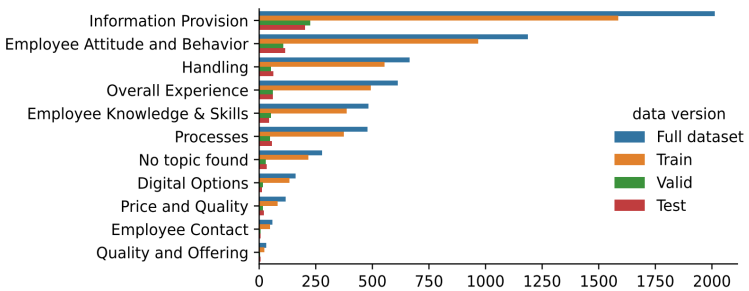



Figure 3.2: Distribution of the topics after the dataset has been split

| Topics | Original | Train | Valid | Test |
|---|---|---|---|---|
| Information Provision | 2,012 | 1,568 | 224 | 202 |
| Employee Attitude and Behavior | 1,186 | 967 | 106 | 113 |
| Handling | 664 | 552 | 51 | 61 |
| Overall Experience | 612 | 492 | 60 | 60 |
| Employee Knowledge & Skills | 481 | 386 | 52 | 43 |
| Processes | 477 | 373 | 48 | 56 |
| No topic found | 277 | 217 | 28 | 32 |
| Digital Options | 159 | 133 | 15 | 11 |
| Price and Quality | 115 | 81 | 15 | 19 |
| Employee Contact | 57 | 46 | 6 | 5 |
| Quality and Offering | 31 | 23 | 3 | 5 |

Table 3.1: Overview after stratified splitting of the original dataset

### 3.1.2 Data Adaptation

Due to the imbalance within the dataset, it has been decided to make use of various data adaptation approaches to the overall dataset and the train dataset in order to evaluate which are the most helpful in the classification task and what the minimum required training samples are. The decisions for the different approaches have been made in close contact with Underlined - debating which methods provide the most advantages to their tools and improve their performance. One approach focuses on the labels by creating new datasets with two new topic schemes, whereas the other two focus more on the feedback sequences, for this the train dataset derived from the original dataset has been adapted. The approaches are explained in detail in the following paragraphs.

#### Merging Topics

It has been decided to merge certain topics in different approaches after a thorough consultation with a representative of Underlined as it is desired to have an increased performance over a vast number of topics. Both merging approaches decreased the number from eleven to eight. Since many of the eleven topics relate to each other and overlap concerning the feedback they contain, the first merging approach is to combine those related topics into one label. This way, the number of labels decreases, and the number of representations within a label increases. The merging approaches are additionally done in order to examine the minimum required number of training samples and the impact it has on the classification task. Before the merging process was finalized, an extensive manual data analysis was necessary. Running first experiments made it obvious that there was some confusion with certain labels. As a next step, the data annotated with said confused labels was looked at more extensively. This resulted in the merging of *Employee Attitude*, *Employee Knowledge  Skills*, and *Employee Contact* into the topic *Employee*. The feedback within those topics is all related to the employees, their skills, their behavior, and the overall contact and can thus be combined into one broader topic. In all three topics, customers describe the friendliness, the helpfulness, as well as the pace of the employees. Another confusion was noticed when looking at *Quality and Offering* and *Price and Quality*. Both topics include feedback about rates, prices, interest, and mortgages. Consequently, it had been decided to merge them and keep the label *Price and Quality* for the topic. The newly created dataset has been split the same way as it is mentioned in Section 3.1.1, Table 3.2 provides an overview of the new labels and their frequency after they were split. Figure 3.3 shows a comparison

between the original and the merged dataset of the overall distribution of the topics. From now on, this dataset will be referred to as *merged* dataset.

| **Topics** | **Total** | **Train** | **Valid** | **Test** |
|---|---|---|---|---|
| Information Provision | 2,012 | 1,568 | 224 | 202 |
| **Employee** | **1,724** | **1,399** | **164** | **161** |
| Handling | 664 | 552 | 51 | 61 |
| Overall Experience | 612 | 492 | 60 | 60 |
| Processes | 477 | 373 | 48 | 56 |
| No topic found | 277 | 217 | 28 | 32 |
| Digital Options | 159 | 133 | 15 | 11 |
| **Price and Quality** | **146** | **104** | **18** | **24** |

Table 3.2: Overview after stratified splitting of the merged dataset

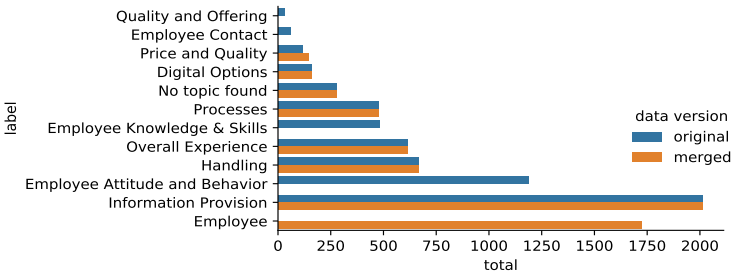



Figure 3.3: Overview frequency after merging labels with overlapping content

The second merging approach focuses on the underrepresented labels. This decision was made after running the first experiments and realizing that some of the classifiers were not at all able to detect underrepresented topics. That means those topics that have a representation below 200 are merged into one label called "Other". In this case, "Quality and Offering", "Employee Contact", "Price and Quality", and "Digital Options" are merged into one, resulting in the distribution shown in Table 3.3 and Figure 3.4. This dataset will be referred to as *other*. Furthermore, it is expected that the performance for classifying this topic will not be outstanding as the content of the topics does not align.

| **Topics** | **Total** | **Train** | **Valid** | **Test** |
|---|---|---|---|---|
| Information Provision | 2,012 | 1,568 | 224 | 202 |
| Employee Attitude and Behavior | 1,186 | 967 | 106 | 113 |
| Handling | 664 | 552 | 51 | 61 |
| Overall Experience | 612 | 492 | 60 | 60 |
| Employee Knowledge & Skills | 481 | 386 | 52 | 43 |
| Processes | 477 | 373 | 48 | 56 |
| **Other** | **362** | **283** | **39** | **40** |
| No topic found | 277 | 217 | 28 | 32 |

Table 3.3: Overview after stratified splitting of the "Other" dataset

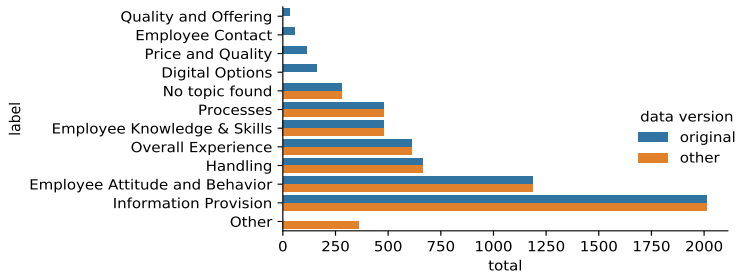



Figure 3.4: Overview frequency after merging underrepresented topics into the topic "Other"

### Sampling

By using sampling methods, the data and its distribution are changed. The sampling method chosen for this project is undersampling and was performed on the train data of the original dataset. Figure 3.5 visualizes the approach of under- and oversampling, however, the focus is only on undersampling.

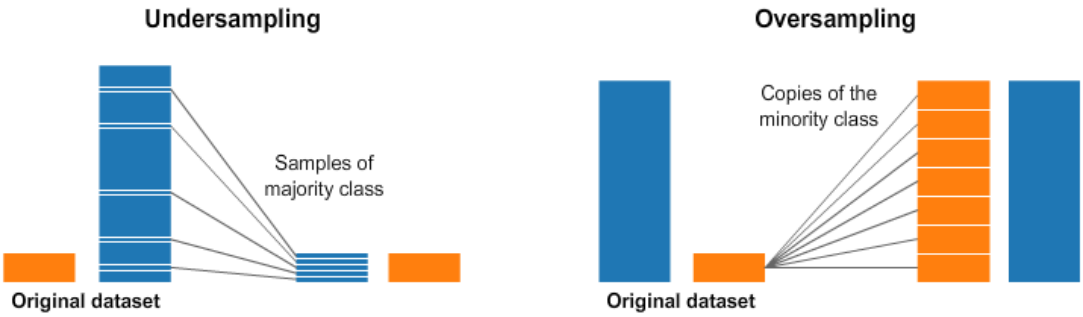



Figure 3.5: Over- and undersampling (Al-Serw, 2021)

Undersampling deletes samples from the majority class, i.e., the class with the highest representation. For this project, the undersampling was done by deleting all the representations until they reached the number of representations of the minority class. Undersampling is implemented specifically to answer the question about the minimum required number of examples per topic and by testing if the amount of training samples of minority classes is sufficient enough for a classifier. It was not implemented to improve the overall results and is handled as a sub-experiment of the classification task. The major downside to this approach is that a lot of sufficient data, valuable information, and examples are being lost in order to comply with the minority class. Table 3.4 showcases the train data after it has been undersampled and the distribution of the topics in the valid and test set.

### Back-Translation

In order to create more variation within the data without changing the feedback sequences too much and to increase the number of training samples, it has been decided to implement back-translation on the training dataset derived from the original dataset.

| Topics | Train | Train U | Valid | Test |
|---|---|---|---|---|
| Information Provision | 1,568 | 23 | 224 | 202 |
| Employee Attitude and Behavior | 967 | 23 | 106 | 113 |
| Handling | 552 | 23 | 51 | 61 |
| Overall Experience | 492 | 23 | 60 | 60 |
| Employee Knowledge & Skills | 386 | 23 | 52 | 43 |
| Processes | 373 | 23 | 48 | 56 |
| No topic found | 217 | 23 | 28 | 32 |
| Digital Options | 133 | 23 | 15 | 11 |
| Price and Quality | 81 | 23 | 15 | 19 |
| Employee Contact | 46 | 23 | 6 | 5 |
| Quality and Offering | 23 | 23 | 3 | 5 |

Table 3.4: Overview after adapting the train data with undersampling (U)

To make use of back-translation, the data has to be translated automatically from the target language into another language and then re-translated back into the target language Sennrich et al. (2015b), as shown in Figure 3.6. The resulting sentence is slightly different from the original without changing the overall content. This method furthermore increases the corpus size. There is also the possibility of translating the target language into several languages and then back-translating it Van Aken et al. (2018) to create more variation and data. Overall, back-translation requires only a small amount of data. Additionally, cross-lingual transfer learning might prove itself to be helpful - by using data from multiple other languages or multi-domain models Chu and Wang (2018). As it can be seen in Figure 3.6, the word *advisor* was back-translated into *consultant* and the word *wanting* into *trying* - showcasing that back-translation creates more variation without making too many changes to the data.

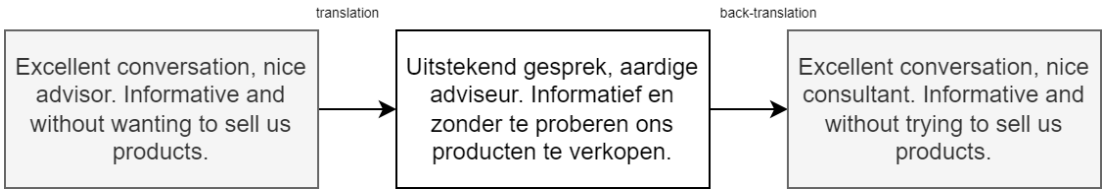



Figure 3.6: Data augmentation using back-translation

Since the original language of the data is Dutch, it has been decided that 10% and 20% of each label are back-translated from English to Dutch and then back to English using a neural machine translator[5], increasing the feedback sequences of the train data from 4,856 to 5,339 and 5,820 respectively. The overall distribution of the topics can be seen in Table 3.5. First the percentage of 10 was chosen to evaluate if a small increase of the train data without creating too much synthetic data has a positive impact on the performances. However, first experiments showed that it only increased the performance for some of the classification algorithms. To see if that performance can be improved even more, 20% per topic was back-translated as a next step.

[5] `https://www.deepl.com/translator`

| Topics | Train | Train 10% | Train 20% | Valid | Test |
|---|---|---|---|---|---|
| Information Provision | 1,568 | 1,744 | 1,901 | 224 | 202 |
| Employee Attitude and Behavior | 967 | 1,064 | 1,161 | 106 | 113 |
| Handling | 552 | 607 | 662 | 51 | 61 |
| Overall Experience | 492 | 541 | 590 | 60 | 60 |
| Employee Knowledge & Skills | 386 | 424 | 462 | 52 | 43 |
| Processes | 373 | 410 | 447 | 48 | 56 |
| No topic found | 217 | 239 | 260 | 28 | 32 |
| Digital Options | 133 | 146 | 159 | 15 | 11 |
| Price and Quality | 81 | 89 | 97 | 15 | 19 |
| Employee Contact | 46 | 50 | 54 | 6 | 5 |
| Quality and Offering | 23 | 25 | 27 | 3 | 5 |

Table 3.5: Frequency per topic after implementing back-translation on 10% and 20% per topic on the train dataset

## 3.2 Automatic Topic Classification

TC is about creating a model that identifies the topic of a feedback statement by matching it to a predefined topic from a variety of topics on which the model has been trained. For this project, the goal is to compare various classification methods on their performance concerning an unbalanced dataset as well as several attempts to balance the dataset. The following sections first introduce the methodology behind the machine learning approaches and the methodology for the transfer learning approaches.

### 3.2.1 Machine Learning

Machine Learning is a field of study that enables computers to learn without having to program them. More specifically, it uses probabilistic and statistical algorithms that learn from an encoded representation of the feedback and its topics. The more data available to teach or train a model, the better.
Concerning this project, the model learns from the labeled data in order to make predictions on new data. This is a supervised machine learning task since the data is already annotated. The opposite, i.e., unlabeled data, is defined as unsupervised machine learning. A combination of both, for instance, when only a small amount of data is labeled, and the rest is not, is called semi-supervised machine learning. To prepare the data for a model, it has to be pre-processed, described in Section 3.2.1 and then converted into vectors using methods like Bag of Words, TF-IDF, or embeddings. Vectors are numerical representations, and their correlation helps the models predict a sequence's topic. Starting from Section 3.2.1, the vectorizing methods of this project are being explained.

#### Pre-processing

In order to structure the data in a way a model can work with and to reduce the noise that appears due to the nature of the feedback data, some pre-processing steps need to be implemented. For TC on user-generated content, typical pre-processing steps are tokenizing and stemming (Kiritchenko et al., 2014), lemmatizing, POS-tagging, and spelling correction (Spasić et al., 2012) or stopword removal, case folding, and normalization (Kusumawati et al., 2019). Since every dataset is different, this project also

requires an individual set of pre-processing steps. The following paragraph summarize the necessary steps taken as well as the motivation behind it.

The first step is to **tokenize** the data, which means dividing the text into individual pieces. By doing so, the text data becomes interpretable for machines once the tokens are vectorized. The remaining pre-processing steps are all implemented on the tokenized data. The next step is removing the names mentioned in the feedback. For the **named entities removal** with a focus on the persons, an adapted name database[6] was used. It had to be adapted as the names mentioned in the feedback were predominantly Dutch and not covered within the list. To be able to find the names, a Dutch Named Entity Recognition[7] was used on the data. This way, the models become more generalizable and do not get distracted by specific names that might frequently appear within the data and have no connection to the actual topic. Additionally, every token is being **lowercased** - simplifying the data even more. As a next step, the **stopwords were removed** since they add no value to the text. Stopwords are commonly used words like "the", "a", or "me" - removing those results in a reduced noise within the feedback sequences. Besides that, **frequent words** derived from a manually created list containing words that are not topic-related are removed for the sake of noise-reduction. Furthermore, **undefined characters, punctuation,** and **digits** are being removed. The undefined characters were due to encoding problems. Here a manually created list comes into the picture as well. As the last step, the feedback is being **lemmatized** on the nouns and verbs in order to get the dictionary form of a token.
Some optional steps were created but not included in this project. However, they can be used in further experiments and make the model more generalizable in a corporate-content. One is the **removal of company-related terms**. If the model is used on other data derived from a different company, it might be helpful to remove terms that are de Volksbank-specific. Additionally, a **spelling correction** was implemented - since the data used for this project was machine translated, it barely contained spelling mistakes, and this pre-processing step was thus not needed.

### Feature Representation: Bag of Words and TF-IDF

A common approach for classification tasks is using Bag of Words (BoW). A BoW, first introduced in 1954 (Harris, 1954), is an unstructured set of words where the position of a word is not taken into consideration but the number of occurrences of the words within a document. Additionally, a Bag of Words does not include grammatical information because of its unstructured nature. The words are associated with vectors that indicate said number of occurrences. The BoW method has the advantage that it is easily implemented and simple to use. However, the semantic meaning of the text data gets lost, as well as the word order, and grammatical information.
In 1972, Term Frequency Inverse Document Frequency (TF-IDF) (Jones, 1972) as a modification of BoW has been introduced. TF represents the frequency of a word within a document, or in this case a feedback. IDF adds a higher weight on words that only occur in a few documents. The IDF makes it possible to discriminate documents from others. It is a numerical statistic with the aim to reflect the importance of a

[6] `https://github.com/smashew/NameDatabases`
[7] `https://huggingface.co/flair/ner-dutch-large`

word in a document, feedback sequences, in a collection of documents, or collection of feedback sequences (Rajaraman and Ullman, 2011). The TF-IDF is vectorized into sparse and long vectors. One vector is a representation of the target word where the dimensions correspond to all the words within the vocabulary.

$$w_{ij} = tf_{ij} idf_i$$

The main difference to BoW is that TF-IDF captures information about the importance of words within the corpus. TF-IDF is easy to compute, but neither captures a word's position within a text nor the semantic information.
After the data has been pre-processed and appears in a tokenized manner, CountVectorizer[8] by sklearn (Pedregosa et al., 2011) is used to implement the BoW representation of the data. It transforms the text into a sparse matrix of n-gram counts. To include more information about the value of the words this matrix is transformed into TF-IDF vectors using TfIdfTransformer[9] also by sklearn (Pedregosa et al., 2011).

**Feature Representation: Embeddings**

An approach to include semantic information and semantic similarities within the vectors was introduced as word2vec (Mikolov et al., 2013). This method maps every word to a high-dimensional vector - also called word embedding and language models. Embeddings are a type of knowledge representation that can capture a text's semantic value by providing an equal representation for words that are close to each other. Language modeling generally refers to the prediction of a text when given a textual context. By using the learned weights in a hidden neural network layer, the model can predict the context words. Compared to a Bag of Words approach, the vectors are small and dense instead of large and sparse. Word2Vec is a prediction-based embedding approach using word-context representation, thus predicting a word given a certain context. It consists of Continuous Bag of Words (CBOW) and Skip-Gram model. CBOW is used to predict the probability of a word given a certain context, whereas a skip-gram predicts the context of a certain word. Embeddings are seen as fast, efficient to train, and readily available. However, word2vec is not able to handle unknown words and cannot interpret them. As a consequence, it uses a random vector. Additionally, it has no morphological understanding and does not recognize words that might have a similar root or words that have one form but multiple meanings. For this project, two pretrained embedding models are used. The first one, BankFin[10] embeddings, is explicitly trained on financial data from the Financial Phrase Bank and several books about comprehensive banking and finance. BankFin embeddings have a dimension of 100. Whereas the second one, GoogleNews[11], which is implemented for evaluation purposes, is trained on parts of the Google News dataset which contains about 100 billion words and has a dimension of 300.

It must be mentioned that except for **tokenization**, there are no pre-processing steps done on the data as it was observed that the ones implemented on Bag of Words

[8] `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html`

[9] `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html`

[10] `https://github.com/sid321axn/bank_fin_embedding`

[11] `https://code.google.com/archive/p/word2vec/`

skew with the performance of embeddings and decrease the contextual information. After the text data has been tokenized, the pre-trained word embeddings models are loaded using gensim (Rehurek and Sojka, 2011). The next step after tokenizing is to retrieve a model's vocabulary and transform the data's tokens into embeddings.

### Classification Algorithms

Besides vectorizing the feedback sequences, it is also necessary to encode the corresponding topics of the sequence from text into numerical labels. This is done with LabelEncoder[12] by sklearn(Pedregosa et al., 2011). Once the data has been completely vectorized, it is ready to train a classifier on it. Generally speaking, in machine learning, classifiers are distinguished into generative and discriminative classifiers. Discriminative classifiers try to learn how to distinguish the topics, whereas a generative classifier focuses on how data is generated in their joint probability distribution. The classifiers Naive Bayes, Logistic Regression, and Support Vector Machine have been chosen for this project. The following paragraphs describe the algorithms in greater detail, including the packages used and which parameters have been adapted to make this project reproducible.

**Naive Bayes** belongs to the probabilistic and generative classifiers. It is called naive because the algorithm assumes that everything in the data works independently. A text is represented as a bag of words, which, as mentioned above, is a set of words that is unordered and disregards the position of the words within a sentence, also called conditional independence assumption (Jurafsky and Manning, 2012). The only thing being kept is the frequency of the word within the data. Naive Bayes was first introduced as the Bayesian inference (Bayes, 1763) and firstly applied to text classification by Mosteller and Wallace (1984). The mathematics behind the Bayes'Theorem are the following:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

In the formula, A and B stand for events. P(A—B) is the probability of event A if event B has occurred; B is also called evidence. P(A) works as the priori of A, i.e., the probability of an event before the model saw any evidence. Lastly, P(B—A) is the probability of event B after the model has seen evidence A. To make it more data science inclusive, the formula behind the classifier is:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

y describes the topic and X the vectorized text sequences. After including the conditional independence and the argmax to find the maximum probability, the formula results in this:

[12] `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html`

$$y = \text{argmax}_y \, P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

Figure 3.7: Naive Bayes formula (Loukas, 2022)

To implement a Naive Bayes algorithm, there are several classifiers available. Some are the Bernoulli Naive Bayes, Categorical Naive Bayes, Complement Naive Bayes, Gaussian Naive Bayes, and Multinomial Naive Bayes. For this project, the latter[13] has been implemented as related work proved its efficiency in text classification. However, it has also been proven that Naives Bayes generally works best for binary classification (Jurafsky and Manning, 2012). Since this project focuses on multi-class classification, Naive Bayes is implemented for evaluation purposes as it is hypothesized that the classifier will provide the lowest results. None of the parameters using the Multinomial NB have been changed, and they are all kept at their default setting.
The advantages of using a Naive Bayes classifier are that the algorithm works fast and well with high-dimensional data. However, with the independence assumption, much valuable information in the text data gets lost.

**Logistic Regression** is used to discover the links between features and predict topics on an s-curved graph, as it can be seen in Figure 3.8, and belongs to the group of discriminative classifiers. Additionally, it is treated as the most important analytical tool in social as well as natural sciences (Jurafsky and Manning, 2012). Because of this superiority, Logistic Regression is, in most cases, the baseline in supervised machine learning algorithms for classification tasks. It can be used for binary and multi-class classification tasks; then it is called multinomial logistic regression, softmax regression, or maxent classifier (Jurafsky and Manning, 2012). The goal for a multinomial logistic regression, according to Jurafsky and Manning (2012), is to find the probability of $y$ being in every potential topic when $y$ is a variable whose range extends two classes. The probability is defined as p(y=c—x). To compute the probability, sigmoid is used as a generalization tool - also described as softmax function. Sigmoid maps numerical value into a value between 0 and 1. To measure how wrong the model is concerning its predictions, a cost function is used - by decreasing this function, the maximum likelihood will increase. To avoid overfitting, regularization is implemented.

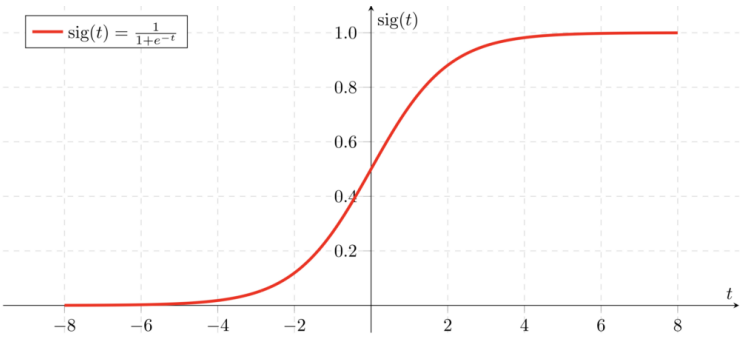



Figure 3.8: S-curved graph of logistic regression, taken from (TowardAI, 2021)

[13] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html

To implement the logistic regression classifier into this project the LogisticRegression classifier[14] from Pedregosa et al. (2011) has been used. Only the maximum number of iterations taken for the solvers to converge were adapted from 100 to 1,000. The remaining parameters have been kept in their default state.

**Support Vector Machine** is the third and last classifier that is being evaluated in this project. It is a linear model used for classification and regression problems, solving linear and non-linear problems using a hyperplane, as shown in Figure 3.9. The overall goal is to find the optimal hyperplane. They are based on the structural risk minimization principle derived from computational learning theory (Joachims, 1998). Support Vectors are the data points that are the closest to the hyperplane.

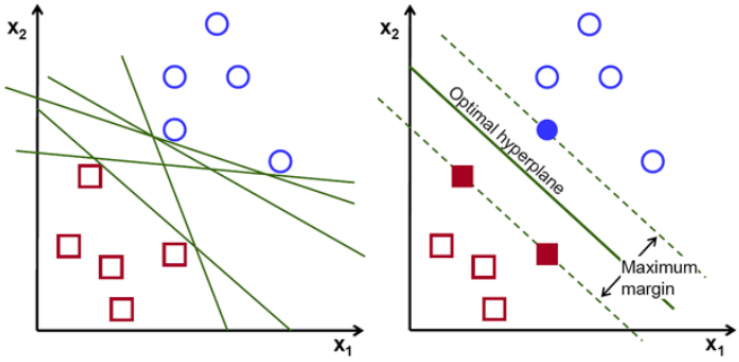



Figure 3.9: Hyperplanes using Support Vector Machine (Gandhi, 2018)

To evaluate the Support Vector Machine's performance for this project, the Linear Support Vector Classification[15] is implemented. The parameters have been kept in their default setting except for the maximum iteration, which has been increased from -1 to 10,000. The advantages of Support Vector Machines, especially for text classification, are their high dimensional input space, the few irrelevant features, the sparsity of the vectors, and the fact that many text categories are linearly separable (Joachims, 1998). On the contrary, SVMs have issues dealing with large datasets and overlapping classes.

### Further Experiments: Feature Engineering

In order to improve the performance of a machine learning model, it is useful to implement various features. This helps the model to classify topics more accurately. Standard features that have been proven to increase the performance in topic classification are the token itself, the surroundings of the tokens in the form of next and previous token but also in the form of n-grams, including a name list, headword of the parsed sentence, and word clusters Toh and Su (2015). Many features are related to the tokens in the form of the frequency of each token, tagging it with their respective Part of Speech (POS), negation detection, and syntactic dependency Liu and Zhang (2012). Some research also includes Named Entity Recognition in the implemented features Menner et al. (2016).

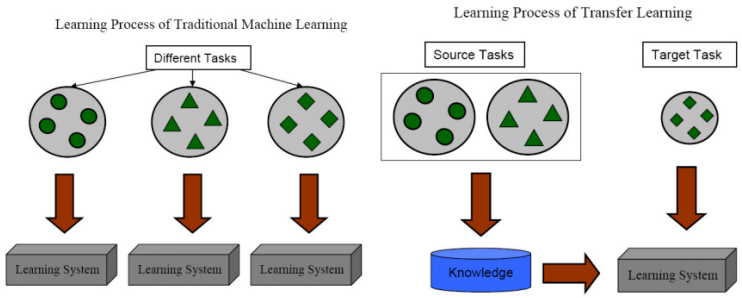
[14] `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`

[15] `https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC1`

Due to the limited time available for this project, it has been decided not to dive into feature engineering. However, future research can implement various feature engineering and feature adaptation steps. This is also why each algorithm's parameters have been kept in their default setting; only the maximum iterations for Logistic Regression have been changed from 100 to 1,000 and for Support Vector Machine from -1 to 10,000. In further experiments, it can be helpful to include more parameter tuning.

### 3.2.2 Transfer Learning and Fine-Tuning

So far, traditional machine learning approaches in classification tasks have achieved great success. However, they all have the same underlying assumption, namely that the training and the test data are taken from the same feature space, and the same distribution (Pan and Yang, 2010). Once that distribution changes, the models must be rebuilt from scratch to work with the new data. In the long run, this is not only expensive but also impossible in the sense of re-collecting the needed (labeled) training data. To solve this issue, transfer learning can be of help. It is about transferring the knowledge across different domains (Zhuang et al., 2020) and makes it possible to reuse pre-trained systems due to the ability to transfer the model's learned knowledge to new tasks and domains. In the field of NLP, transfer learning has recently been introduced and has significantly improved many tasks. Especially in this field, sequential transfer learning is used, which starts with pre-training a model where it learns general representations on a specific source task or a domain and is followed by the adaptation where the learned knowledge of the model is applied to a target task or domain (Ruder et al., 2019). Figure 3.10 shows a comparison of traditional machine learning approaches and the transfer learning process.




(a) Traditional Machine Learning (b) Transfer Learning

Figure 3.10: Comparison Process Traditional Machine Learning and Transfer Learning

Transfer learning is used to fine-tune already trained models for task-specific features. The model has been pre-trained on a general task and is then fine-tuned by training only specific layers with a different target task. In this project, transformer-based models have been chosen, i.e., they have been pre-trained on general tasks and will be fine-tuned on the data from the banking domain.
Transformers have been built up on word2Vec and Recurrent Neural Networks (RNN) and are currently the state-of-the-art architecture for language modeling. Language models predict the following word by considering the previous words of a sequence as the context (Malte and Ratadiya, 2019). Word2Vec has been explained in greater detail in Section 3.2.1. RNNs (Jordan, 1986) improve on word embeddings by including the word context. For this, various approaches have been created, starting with Long-

Short-Term memory (LSTM) (Hochreiter and Schmidhuber, 1997) capturing long-term dependencies. In the same year, Bidirectional RNN (Schuster and Paliwal, 1997), a model that captures the dependencies left-to-right and right-to-left, was also published. Finally, the encoder-decoder RNN (Cho et al., 2014) has been proposed recently, which focuses on creating document embeddings with encoding and decoding those back into text. Even though RNNs provide a solution for the contextualization issue, they come with certain disadvantages. First, they are slow in their performance, and secondly, the vanishing gradient, i.e., the model "forgets" important aspects of the sentence. Another issue is that most RNN-based models work unidirectional, meaning only from one direction to the other. This causes the model to miss certain contexts. Introduced in 2017, the current state-of-the-art architecture Transformers (Vaswani et al., 2017) addresses the issue of unidirectionality. It is an encoder-decoder model that makes use of attention to improve the computation of embeddings and the alignment of output and input. This section first introduces the architecture of Transformers, then presents the three pre-trained language models, BERT, RoBERTa, and DistilBERT that have been chosen for this project.

Before describing the models one by one, the overall architecture of transformers is described in the following section based on Alammar (2018b). Figure 3.11 provides an overview of the architecture including additional explanations.

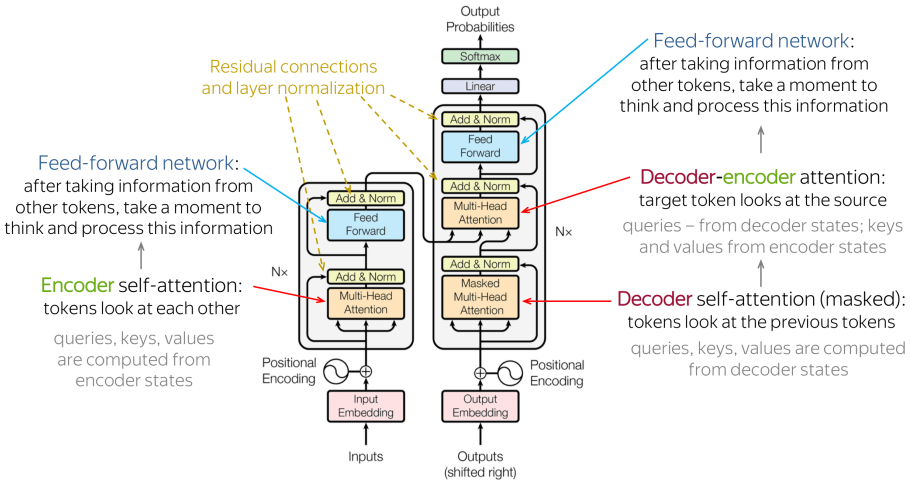



Figure 3.11: Transformer Architecture (Voita, 2022)

In its basis, a transformer works with encoding and decoding components that are connected. The encoding component consists of a stack of encoders. In the original paper Vaswani et al. (2017) there are six encoders within the stack - other variations, of course, can also be applied - as well as six decoding components within the decoder stack. The encoders contain two sub-layers. First, an encoder self-attention which helps the encoder observe the other words in a sentence for clues while it encodes one word. The output is then independently fed into a feed-forward neural network, the second sub-layer of an encoder. The decoder component consists of three sub-layers, self-attention, encoder-decoder attention, and a feed-forward neural network. An encoder-decoder attention layer is implemented in order for the decoder to focus on parts of an input sentence with higher relevance.

In greater detail, the architecture begins with vectorizing each word by using an embedding algorithm introduced at the beginning of the encoder. Transformers include a positional encoding after the embedding, i.e., a way to preserve the order of the words of the input sentences. The positional encoding comes in the form of a vector added to each input embedding. After that, it is fed through the first encoder and its two layers; self-attention and feed-forward neural network. The encoder receives a list of vectors in a size that can be set manually - it often is set to 512. This list is being processed and each encoder passes its processed list of vectors to the next encoder and its two sub-layers, i.e., every word goes through its individual path within the encoder.
As mentioned above, self-attention helps the model understand the other relevant words within the input in relation to the one word being processed. Simply put, each token looks at the other tokens within the sentence, collects the context, and thus updates the representation of the 'self' (Voita, 2022). Three vectors need to be calculated: query vector, key vector, and value vector. They are created by multiplying the embedding with three matrices that have been trained in the training phase. The vectors are all of a similar structure. A vector represents each word within a sequence. The vectors are smaller in size compared to the embedding vector, which helps make the multiheaded attention more constant. The query vector is asking for information, the key vector says it has information, and the value vector provides the information (Voita, 2022). As a next step, a score is being calculated that indicates how much focus should be placed on the other parts of an input sentence while encoding one individual word of the sentence. It is calculated with the dot product of the query and key vector of the individual word. Furthermore, the following steps focus on dividing the scores by the square root of the key vectors' dimension to have a more stable gradient. The result undergoes a softmax operation to normalize the scores. In the end, the scores are favorable and, when added together, result in 1. This softmax score decides how likely each word in this exact position will be expressed - the word at this position will provide the highest softmax score. In the next step, each value vector is multiplied with the softmax score to exclude irrelevant words and to keep the values of the word intact. This is followed by summing up the weighted value vectors to receive the output of the self-attention layer for the individual word. The resulting vector is fed to the feed-forward neural network. To speed up the process, the self-attention is calculated using matrices, as shown in Figure 3.12. The first step here is to calculate the matrices for query, key, and value by packing the embeddings into a matrix and multiplying it with the weight matrices that have been trained. As a next step, the output of the self-attention layer can already be calculated using the softmax.

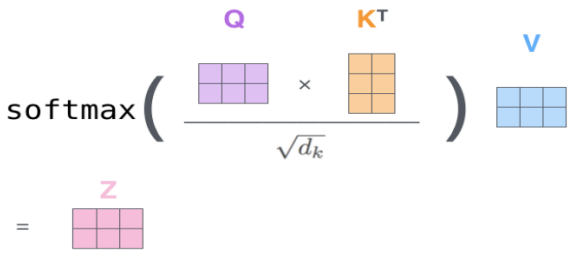



Figure 3.12: Self-attention calculation in matrix form (Alammar, 2018b)

In order to improve the performance of the attention layers, the multi-headed attention mechanism is added. It enhances the performance by expanding the model's

capacity to concentrate on various positions and adding representation subspaces. The multi-head attention mechanism calculates the self-attention several times with the different weight matrices. The number of calculations is determined by calculating *Query Size = Embedding Size/Number of heads* (Doshi, 2021). This results in n matrices that are concatenated and multiplied with an additional weights matrix in order for the feed-forward neural network to work with it. Figure 3.13 provides an overview of the process of multi-headed self-attention.

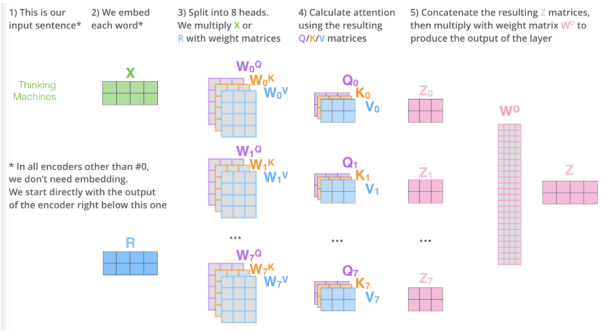



Figure 3.13: Process multi-headed self-attention (Alammar, 2018b)

The attention layer requires several hyperparameters. Starting with the embedding size, so the width of an embedding vector, followed by the query size, the number of attention heads, and lastly, the batch size (Doshi, 2021).

Diving deeper into the structure of the encoder components as well as the decoder components, the yellow boxes in Figure 3.11 describe "Add  Norm". *Add* stands for the residual connections around both sub-layers. They improve the performance by easing the gradient flow and allowing the stacking of many layers (Voita, 2022). The *Norm* represents the layer-normalization (Ba et al., 2016). It normalizes vector representation to control the flow to the next layer and improves convergence stability (Voita, 2022). So far, the focus has been more on the encoding component. The decoding one works in mostly the same way. Once the output of the encoder is transformed into a set of attention vectors, they are fed into the decoder-encoder attention layer within the decoder component. This layer aids the overall decoder in focusing on the correct placements in the input sequence. This step is repeated until a specific symbol has been reached, indicating that the decoder component has completed the output. This output is then fed to the bottom decoder, embeds it, adds a positional encoding and inputs it into the next decoder component. Compared to the encoder, the decoder differs concerning the self-attention layer since it only focuses on earlier positions of the output sequence. This focus is achieved by masking future positions before the softmax takes place. The encoder-decoder attention layer has the exact mechanisms as the multi-headed self-attention with the exception that here a queries matrix is created with the layer below it and then takes the key and value matrices from the output of the encoder.

The final linear layer and the softmax layer following the decoder mechanism are used to turn the numerical representation into a word. The decoder stack returns a vector of floats which is then fed into the linear layer. This layer is a fully connected neural network that turns the output of the decoder stack into a large vector called logits vector. Those logits are turned into probabilities by the softmax layer. The

probabilities are all positive and, if added together, would result in 1.0. As a next step, the highest probability is chosen and the corresponding word is presented.

### Fine-Tuning pre-trained Transformer Models

Training a transformer model from scratch is time-consuming and requires massive data for the model to learn the representations. Luckily, pre-trained language models (PTM) exist and can be fine-tuned on downstream tasks, like TC, using annotated data. For transformer models, it has been decided to fine-tune BERT, RoBERTa, and DistilBERT. The overall comparison can be seen in Table 3.6 providing an insight into their size, the data they have been trained on, and the methods they use. Moreover, the following sections describe their differences and their alignments in greater detail.

| | BERT (Devlin et al., 2018) | RoBERTa (Liu et al., 2019) | DistilBERT (Sanh et al., 2019) |
|---|---|---|---|
| Size (in millions) | Base: 110, Large: 340 | Base: 110, Large: 340 | Base: 66 |
| Data | 16GB (Wikipedia, Book Corpus), 3.3 Billion Words | 160GB (BERT + 114GB additional) | 16GB (Wikipedia, Book Corpus), 3.3 Billion Words |
| Method | Bidirectional Transformer with MLM and NSP | BERT without Next Sentence Prediction | BERT Distillation |

Table 3.6: Overview Transformers, adapted from (Khan, 2021)

**BERT** stands for *Bidirectional Encoder Representations from Transformers* and has been introduced by Devlin et al. (2018) from Google. A simplified version of its architecture can be seen in Figure 3.14.

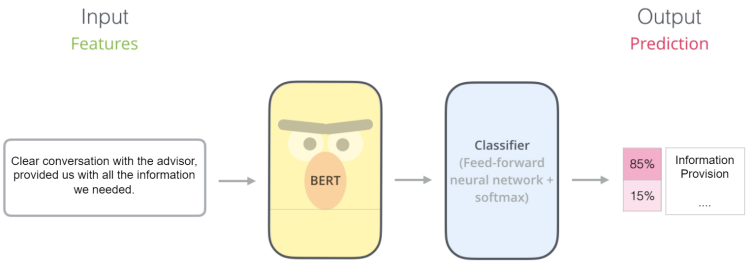



Figure 3.14: Basic procedure BERT, adapted from Alammar (2018a)

It has been trained on 16GB of uncompressed text data, including books and the English Wikipedia. Instead of using the encoder-decoder architecture, it only employs a bidirectional encoder in order to learn the contextual representations of words, optimized with a Masked Language Model (MLM) and a Next Sentence Prediction (NSP) (Hadi and Fard, 2021). Concerning the MLM, the model replaces 15% of all tokens with a masked token - written like [MASK] - before the model is being trained. The model then has to predict the masked word based on the context it has learned concerning the un-masked words. More specifically, 80% of the masked tokens were marked as [MASK], 10% was replaced with another random word, and the remaining 10% kept

the word as it is (Malte and Ratadiya, 2019). Through MLM, the model learns the relation between the tokens within the sequences.
The NSP is implemented by taking sentence pairs as input for the model so it can learn to predict the correctness and wrongness of a pair. An input sequence looks like this *[CLS] <Sentence A >[SEP] <Sentence B >[SEP].* [CLS] is the first input token and stands for *Classification.* It is used to obtain a fixed vector representation and [SEP] is used to separate the two input sequences (Malte and Ratadiya, 2019). This step helps the model to learn the long-distance dependencies.
For BERT, two versions were trained by Devlin et al. (2018). The small-sized BERTBASE, which is implemented in this project, and the big-sized BERTLARGE. The BERTBASE has 12 encoder stacks and 110 million parameters, whereas BERTLARGE has 24 encoder stacks and 340 million parameters. BERTBASE was trained on 16 TPU chips and BERTLARGE on 64 TPU chips - both training procedures took four days (Malte and Ratadiya, 2019). Compared to the initial transformer architecture, BERTBASE and BERTLARGE also have more extensive feed-forward neural networks; BERTBASE has 768 and BERTLARGE has 1,024 hidden units, and more attention heads; BERTBASE has 12 and BERTLARGE has 16. Since BERTLARGE is not only computationally expensive but also consumes way more memory, it has been decided to work with BERTBASE. It is considered to be the state-of-the-art of 11 NLP tasks (Hadi and Fard, 2021).
For BERT, the maximum length of the input sequence has to be restricted, this is usually put to 512 tokens, a dropout value of 0.1 functions as regularization, and instead of a RElu as an activation function, a GELU function is used (Malte and Ratadiya, 2019). GELU stands for Gaussian Error Linear units, and compared to ReLU and eLU improves the model's performance.
In order to fine-tune BERT, only a small dataset is needed and the model needs less processing power (Malte and Ratadiya, 2019). BERT has been made publicly available, resulting in many new pre-trained models based on BERTs architecture.

**RoBERTa** is short for *Robustly optimized BERT approach* and has been publicized as a replication study by Liu et al. (2019) from Meta. It has been pretrained with 1,000% more data and computer power. 160GB of text data was derived from CommonCrawl News dataset, Web text, and Stories from CommonCrawl (Khan, 2021). Additionally, they increased the mini-batch sizes and the training time to train the model, as well as, including data that consists of longer sequences (Hadi and Fard, 2021). Furthermore, another step that has been taken is the removal of Next Sentence Prediction; henceforth, RoBERTa is only making use of Masked Language Model. Liu et al. (2019) showed that the performance is either not affected or is actually improved by removing it. The model outperforms BERT and other models in many different NLP tasks.

**DistilBERT** brought to the public by Sanh et al. (2019) is a distilled version of BERT that retains 97% of BERT's performance but only makes use of 50% of the number of parameters (Khan, 2021). Knowledge Distillation is about training a smaller student while being supervised through a larger and more accurate teacher model (Singh and Mahmood, 2021). DistilBERT was created through distillation of pre-trained BERT in order to mitigate computational costs due to the large size of the other models that require several GPUs to load, which in the long run is costly and

resource prohibitive (Singh and Mahmood, 2021). It reduces the BERT size by 40% and is 60% faster while requiring lesser training cost (Singh and Mahmood, 2021). Due to its pace and so far promising results, it has been decided to implement it next to RoBERTa and BERT to evaluate if it is a beneficial model for Underlined.

The fine-tuning approach was adapted from a thesis project[16] by Catanese (2021) who adapted the approach from Mihaila (2020). The three different transformer-based models that are used in this project are available on huggingface[17]. Mihaila (2020) is making use of AutoClasses[18] which is able to guess the model's configuration, tokenizer, and architecture, simply by including the name of the model. In order to fine-tune the models, PyTorch (Paszke et al., 2019) is being used. Mihaila (2020) project had to be adapted because it focuses on a binary classification task, whereas this project focuses on multi-class classification.
The setup for the parameters for all three transformer-based models is the same in order to be able to evaluate the models on similar matters. The number of training epochs is **4**, the size of the batches is **32**, and the maximum length is **128**[19]. As a learning rate, we chose **3e-5** and AdamW for the Optimizer. The specific models that were implemented were BERT-base-uncased, DistilBERT-base-uncased, and RoBERTa-base.

The first step of the model is to pre-process the data. Therefore the text data is fed into the model and then tokenized using the built-in tokenizer. This creates a WordPiece or a Byte Pair Encoding (BPE) representation. For BERT and DistilBERT, the tokens are being lowercased. The BPE is used to split words into sequences of characters that contain special tokens to showcase the beginning or the end of a sequence (Sennrich et al., 2015a). These representations are padded to an equal length of 128. As a next step, the topics are encoded and the batches are constructed.
After completing the pre-processing steps, the fine-tuning begins. The model learns the relation between the input features and their respective topic in this process. To achieve this, a vector space is assigned to a token in combination with other tokens with the same context. The model then learns the association between the space and the topic and returns the probability of a sequence to be classified as a specific topic. As a next step, this process is validated by implementing the validation data set in each epoch. This way, the error is minimized, and the weights are adjusted until the model reaches an efficient performance.

Since the data cannot be shared with third-parties, using virtual machines such as Google Colaboratory[20] that provide access to GPUs and TPUs which are incredibly helpful for running transformer-based models was not an option. For this experiment, a 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 1.38 GHz with a 8GB RAM was used.

[16] `https://github.com/cltl-students/catanese_gabriele_text_mining_thesis`
[17] `https://huggingface.co/`
[18] `https://huggingface.co/docs/transformers/model_doc/auto`
[19] The batch size and maximum length were taken from `https://github.com/google-research/bert#out-of-memory-issues`
[20] `https://colab.research.google.com/`

# Chapter 4

# Results

In order to evaluate the performance of the different classification models, we use the macro-averaged precision, recall, and F1-score. The scores are explained in the next paragraph and, if not specified otherwise, based on the explanation provided by Jurafsky and Manning (2012).
Precision and recall are calculated using True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). TP and TN describe the correct predictions. Prediction is seen as TP when the gold labels and the predictions align, and TN is when the observation is correctly predicted as negative. FP describe wrong predictions by the classifier that do not align with the gold labels and FN the opposite.

Precision describes the percentage of items detected by a system, and the predictions match the gold labels. For instance, the gold labels state that one feedback is "Employee Contact" and the classifier predicts this label as well. It is calculated by taking the number of correctly predicted topics, True Positives (TP), and dividing them by the sum of the TP and the wrongly predicted, False Positives (FP).

$$P = \frac{TP}{TP + FP}$$

Recall is defined as measuring the percentage of items present in the input and correctly identified by the model. It is calculated by dividing the TP by the sum of TP and FN.

$$R = \frac{TP}{TP + FN}$$

The F1-score is the harmonic mean of precision and recall.

$$F\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

The macro average computes the performance for each class and then averages over the classes - meaning it uses the arithmetic mean of all F1-scores.
The evaluation of the classification task was performed using the test set. The test set contains 607 examples, i.e, 10% of the original data set. The following sections present the results of the original data and the various experimental setups. The focus of the evaluation scores will be the macro averaged precision, recall, and F1-score. The classification report was created using scikit-learn[1] (Pedregosa et al., 2011).

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html`

## 4.1 Results

The scores in the following sections are represented in relation to the research questions. First, summarizing the results of the different classification algorithms on the original dataset, then the data augmentation techniques, and lastly, the results of the different data adaptation methods. Due to the size of the results, the complete tables can be found in the Appendix. The following sections will provide an overview of the most important results, presenting the macro averaged precision (p), recall (r), and F1-score (f).

### 4.1.1 Results: Original Dataset

Table 4.1 provides an overview of the different classifiers and their respective performances on the original dataset. Overall the BoW TF-IDF approach with an SVM algorithm outperforms the embedding approach, as well as the transfer learning algorithms with a macro averaged F1-score of 0.537. The other classifiers with the BoW TF-IDF representation scored relatively lower; Naive Bayes has an F1-score of 0.250 and Logistic Regression 0.455.
Concerning the embeddings as feature representation, the performance of the SVM with the GoogleNews embeddings is comparable to the best-performing model with an F1-score of 0.530, only being lower by 0.007. The other embedding approaches scored between 0.293 and 0.367. However, the transformer-based models did provide comparable scores to the best-performing classifier, providing F1-scores between 0.517 and 0.530.

| **Classifier** | **Feature Representation** | *m avg* | | |
|---|---|---|---|---|
| | | *p* | *r* | *f* |
| **NB** | BoW + TF-IDF | 0.897 | 0.241 | 0.250 |
| **LogReg** | BoW + TF-IDF | 0.641 | 0.424 | 0.455 |
| **SVM** | BoW + TF-IDF | 0.566 | 0.520 | **0.537** |
| **LogReg** | Embeddings: BankFin | 0.565 | 0.299 | 0.331 |
| **SVM** | Embeddings: BankFin | 0.460 | 0.279 | 0.293 |
| **LogReg** | Embeddings: GoogleNews | 0.671 | 0.324 | 0.367 |
| **SVM** | Embeddings: GoogleNews | 0.656 | 0.343 | 0.530 |
| **BERT** | | 0.525 | 0.516 | 0.517 |
| **RoBERTa** | | 0.530 | 0.541 | 0.528 |
| **DistilBERT** | | 0.537 | 0.523 | 0.525 |

Table 4.1: Results TC on the original dataset concerning macro-averaged precision (p), recall (r), and F1-score (f)

It can be concluded that on this dataset, the BoW TF-IDF with an SVM classifier provides the superior performance but is closely followed by the state-of-the-art transformer-based models and an SVM with GoogleNews embeddings as a feature representation. It can be assumed that BoW TF-IDF might outperform embeddings because of the nature of the dataset with domain-specific vocabulary as well as its relatively small size. Overall, the GoogleNews embedding model outperformed the BankFin embeddings. This can be due to the fact that the BankFin embeddings have not been trained on a big enough corpus that mainly includes more colloquial language surrounding the banking domain, i.e., the embeddings might have been too domain-specific and

not suitable for user-generated feedback.

Furthermore, the classifiers were able to predict the topics to some extent correctly. However, the performance overall is still relatively low. As a next step, data augmentation techniques have been implemented in order to improve the performance by increasing the size of the training samples.

### 4.1.2 Results: Data Augmentation

For the data augmentation, it has been decided to randomly back-translate 10% and 20% of each topic to create more variety and increase the sample size.
The best performing classifier concerning the first data augmentation approach, back-translating 10% of each topic, as shown in Table 4.2, is again SVM with BoW TF-IDF with an F1-score of 0.527. However, it has to be mentioned that the performance dropped from 0.537 to 0.527. That can also be observed for Naive Bayes, dropping from 0.250 to 0.248 and still being the lowest-performing classifier. Also, Logistic Regression with the BankFin embeddings, Logistic Regression with the GoogleNews embeddings, RoBERTa, and DistilBERT experienced a decrease in their performance. However, some classifiers experienced an increase. Logistic Regression with the BoW representation rose from 0.455 to 0.469, SVM with the BankFin embeddings from 0.293 to 0.297, and BERT from 0.517 to 0.523, making it the second-best performing classifier.

| **Classifier** | **Feature Representation** | *m avg* | | |
|---|---|---|---|---|
| | | *p* | *r* | *f* |
| **NB** | BoW + TF-IDF | 0.861 | 0.240 | 0.248 |
| **LogReg** | BoW + TF-IDF | 0.640 | 0.437 | 0.469 |
| **SVM** | BoW + TF-IDF | 0.550 | 0.516 | **0.527** |
| **LogReg** | Embeddings: BankFin | 0.557 | 0.275 | 0.295 |
| **SVM** | Embeddings: BankFin | 0.469 | 0.283 | 0.297 |
| **LogReg** | Embeddings: GoogleNews | 0.654 | 0.322 | 0.364 |
| **SVM** | Embeddings: GoogleNews | 0.597 | 0.343 | 0.381 |
| **BERT** | | 0.523 | 0.522 | 0.523 |
| **RoBERTa** | | 0.513 | 0.525 | 0.513 |
| **DistilBERT** | | 0.525 | 0.508 | 0.515 |

Table 4.2: Results TC on the back-translated 10% train dataset

Back-translating only 10% per topic caused improvements and drawbacks for some classifiers, the overall best F1-score falling from 0.537 on the original dataset to 0.527. The same phenomenon can be observed when examining the results of the 20% back-translation in Table 4.3. Compared to the performance after being trained and fine-tuned on the original dataset, the performance for the majority of the classification algorithms dropped. That is the case for SVM with BoW TF-IDF, Logistic Regression with the BankFin embeddings, SVM with the GoogleNews embeddings, and all three transformer-based models. However, the performance improved for Naive Bayes from 0.250 to 0.252, for Logistic Regression and BoW from 0.455 to 0.467, for SVM with BankFin embeddings from 0.293 to 0.298, and the Logistic Regression with GoogleNews embeddings from 0.367 to 0.378.

However, compared to the results of the 10% back-translation, some of the classifiers improved again. Nevertheless, the scores mostly do not reach the scores after being trained and fine-tuned on the original dataset. BERT provides the best performance with an F1-score of 0.525, which comes relatively close to the best-performing classifier on the original dataset with a score of 0.527.

| **Classifier** | **Feature Representation** | *m avg* | | |
|---|---|---|---|---|
| | | *p* | *r* | *f* |
| **NB** | BoW + TF-IDF | 0.863 | 0.244 | 0.252 |
| **LogReg** | BoW + TF-IDF | 0.718 | 0.441 | 0.467 |
| **SVM** | BoW + TF-IDF | 0.541 | 0.510 | 0.520 |
| **LogReg** | Embeddings: BankFin | 0.538 | 0.270 | 0.287 |
| **SVM** | Embeddings: BankFin | 0.555 | 0.284 | 0.298 |
| **LogReg** | Embeddings: GoogleNews | 0.667 | 0.322 | 0.378 |
| **SVM** | Embeddings: GoogleNews | 0.597 | 0.350 | 0.389 |
| **BERT** | | 0.526 | 0.531 | **0.525** |
| **RoBERTa** | | 0.522 | 0.536 | 0.521 |
| **DistilBERT** | | 0.528 | 0.512 | 0.516 |

Table 4.3: Results TC on the back-translated 20% train dataset

The data augmentation partially helped the performance but mostly showcased issues in its implementation. By simply looking at the scores, it cannot be labeled as a promising data augmentation technique which is why the predictions will be analyzed more thoroughly at a later step. However, there is room for improvement Since the back-translated data was chosen randomly, this might have impacted the performance as not all feedback statements provide the same level of quality.

This data augmentation was done in order to improve the overall performance of the classifier; however, other techniques have been implemented as well to examine how the performance can be improved for the tools of Underlined.

### 4.1.3 Results: Merging Topics

To improve the data and especially the topic distribution, it has been decided to merge several topics. After a deep examination of the data and consulting with representatives of Underlined, it was decided to implement merging topics for an improvement of the results due to merging similar classes.
In the first approach, topics with overlapping content were merged in order to decrease noise and reduce the number of topics. It was additionally done to investigate how many training samples are required for a classifier to provide reasonable results. Table 4.4 provides an overview of the performance of the different classifiers.

| **Classifier** | **Feature Representation** | *m avg* | | |
|---|---|---|---|---|
| | | *p* | *r* | *f* |
| **NB** | BoW + TF-IDF | 0.824 | 0.335 | 0.348 |
| **LogReg** | BoW + TF-IDF | 0.696 | 0.534 | 0.573 |
| **SVM** | BoW + TF-IDF | 0.637 | 0.589 | 0.605 |
| **LogReg** | Embeddings: BankFin | 0.495 | 0.352 | 0.378 |
| **SVM** | Embeddings: BankFin | 0.442 | 0.348 | 0.362 |
| **LogReg** | Embeddings: GoogleNews | 0.620 | 0.394 | 0.437 |
| **SVM** | Embeddings: GoogleNews | 0.668 | 0.409 | 0.451 |
| **BERT** | | 0.661 | 0.669 | 0.661 |
| **RoBERTa** | | 0.675 | 0.670 | **0.668** |
| **DistilBERT** | | 0.656 | 0.628 | 0.637 |

Table 4.4: Results TC on the merged dataset

RoBERTa outperforms the other classification algorithms with an F1-score of 0.668, followed by BERT with 0.661, and DistilBERT with 0.637. The merging of the topics overall increased the performance, indicating that merging topics with overlapping content might be sufficient, and a smaller amount of topics is possible for this dataset and likewise datasets.

Another topic adaptation technique concerned merging underrepresented classes into one topic called "Other". Table 4.5 provides an overview of the results, and it can be seen that RoBERTa outperformed the other classification algorithms with an F1-score of 0.670. Next in rank comes BERT with a score of 0.667, followed by DistilBERT with a score of 0.651. Since this dataset has a different distribution of topics, it cannot directly be compared to the performance of the classifiers trained on the original or the merged dataset. However, it can be observed that with a lower number of classes, the performance of the transformer-based models rises, and the embedded models still provide the lowest results in total. Naive Bayes keeps scoring the lowest F1-score with a score of 0.344.

| **Classifier** | **Feature Representation** | *m avg* | | |
|---|---|---|---|---|
| | | *p* | *r* | *f* |
| **NB** | BoW + TF-IDF | 0.817 | 0.331 | 0.344 |
| **LogReg** | BoW + TF-IDF | 0.652 | 0.552 | 0.577 |
| **SVM** | BoW + TF-IDF | 0.655 | 0.613 | 0.627 |
| **LogReg** | Embeddings: BankFin | 0.473 | 0.359 | 0.380 |
| **SVM** | Embeddings: BankFin | 0.467 | 0.356 | 0.373 |
| **LogReg** | Embeddings: GoogleNews | 0.623 | 0.422 | 0.465 |
| **SVM** | Embeddings: GoogleNews | 0.602 | 0.439 | 0.479 |
| **BERT** | | 0.678 | 0.661 | 0.667 |
| **RoBERTa** | | 0.685 | 0.664 | **0.670** |
| **DistilBERT** | | 0.667 | 0.640 | 0.651 |

Table 4.5: Results TC on the "Other" dataset

Overall, it can be observed that the merging approaches, obviously, improve the classification task due to the lower number of topics. Both merging approaches decreased the number from eleven to eight. The strategy, however, is primarily beneficial for the transformers, as, for instance, the performance of RoBERTa rises from 0.528 to 0.668,

which is more than 0.1. The same can also be observed for the other transformer-based models. Concerning traditional machine learning classifiers, the merging still helped to improve the performance, although not to the same extent. It indicates that SVM with a BoW TF-IDF representation seems to be performing better on the original setup as it is especially able to capture underrepresented topics.
After the topics have been merged, the question arises of how many samples are actually needed as well as sufficient in the training data for the classifier to be able to predict the topics correctly. The following section focuses on a data reduction experiment, especially conducted for this question.

### 4.1.4 Results: Data Reduction

To perform the data reduction, the topics were all undersampled to the minority class, i.e., to a total of 23 samples per topic. Table 4.6 shows how the classification algorithms performed after being trained and fine-tuned on the undersampled dataset.

| **Classifier** | **Feature Representation** | *m avg* | | |
|---|---|---|---|---|
| | | *p* | *r* | *f* |
| **NB** | BoW + TF-IDF | 0.318 | 0.358 | 0.305 |
| **LogReg** | BoW + TF-IDF | 0.324 | 0.377 | **0.342** |
| **SVM** | BoW + TF-IDF | 0.334 | 0.398 | 0.333 |
| **LogReg** | Embeddings: BankFin | 0.246 | 0.247 | 0.215 |
| **SVM** | Embeddings: BankFin | 0.237 | 0.265 | 0.217 |
| **LogReg** | Embeddings: GoogleNews | 0.339 | 0.333 | 0.277 |
| **SVM** | Embeddings: GoogleNews | 0.305 | 0.316 | 0.266 |
| **BERT** | | 0.089 | 0.149 | 0.073 |
| **RoBERTa** | | 0.064 | 0.102 | 0.06 |
| **DistilBERT** | | 0.288 | 0.205 | 0.093 |

Table 4.6: Results TC on the undersampled train dataset

Logistic Regression with BoW TF-IDF outperformed the other models with an F1-score of 0.342. However, the other BoW TF-IDF approaches scored around 0.3 as well, and the embedded ones were around 0.2. The transformer-based models provided the lowest performance, with F1-scores between 0.060 to 0.093.
Compared to the original dataset, the scores for most classification algorithms decreased. However, Naive Bayes had an improvement of 0.055. Since the dataset is so highly undersampled, it was not expected that the classifiers would provide satisfactory results - however, the results will be analyzed more deeply further on.

### 4.1.5 Results: Concluding Remarks

Before the results are analyzed in greater detail, a short interim conclusion can be drawn. For the original dataset, the best approach was to use BoW with TF-IDF and a Support Vector Machine scoring an F1-score of 0.537. When comparing the performance of all experimental setups, the best performing algorithm is RoBERTa after the underrepresented topics have been combined into one topic, providing an F1-score of 0.670. However, this was to be expected due to the reduction of topics. In most cases, Support Vector Machine outperforms the other classifiers. The superiority of the BoW TF-IDF approach can be due to its keyword-based mechanism and the

relatively small dataset, domain-specific vocabulary, and sentence structure. The pre-trained embedding models provided a weaker performance. Overall, the GoogleNews embedding model worked better than the BankFin embeddings. This might be because the BankFin embeddings have not been trained on a large enough corpus but on a domain-specific one that does not include colloquial language.

## 4.2 Evaluation of the Results - Error Analysis

As mentioned above, this section analyzes the predictions in greater detail. At first, the best-performing classifier trained on the original dataset will be examined, followed by an evaluation of the results of each adapted dataset concerning the classifier with the highest performance. For this, a confusion matrix illustrates the alignment between the gold labels and the predictions made by the classification algorithm. The examples in the following sections are made-up to avoid publishing the restricted data. Additionally, notions for future research are mentioned.

### 4.2.1 Analysis: Topic Classification on the Original Dataset

Support Vector Machine with a BoW and TF-IDF feature representation outperformed the other classification algorithms with an F1-score of 0.537, identifying 427 out of 607 feedback statements. The following confusion matrix in Figure 4.1 gives an insight into the classification performance.

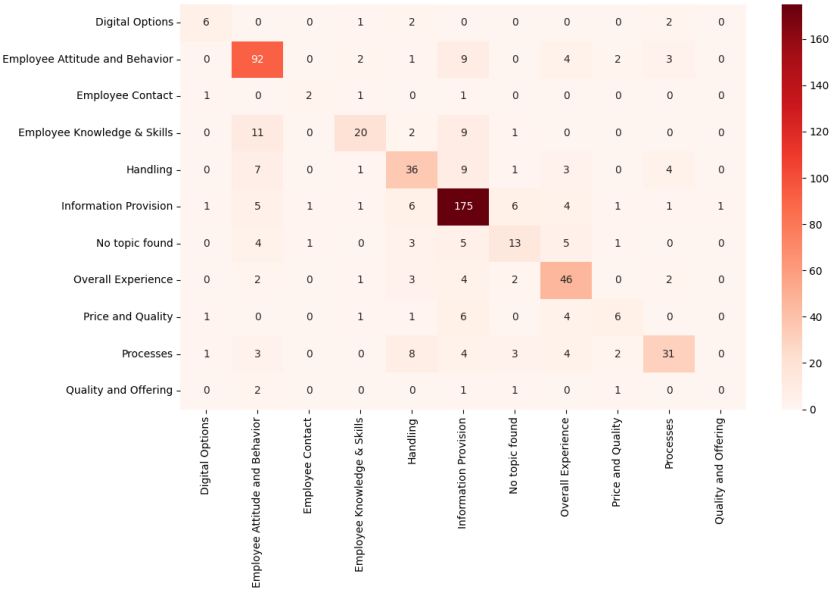



Figure 4.1: Confusion Matrix SVM BoW TF-IDF Original Dataset

SVM was able to correctly predict all topics, except *Quality and Offering*, at least to some extent. The undetected topic was misclassified with the topics *Employee Attitude and Behavior, Information Provision, No topic found,* and *Price and Quality.* As mentioned in Section 3.1, the feedback should be about offers made from de Volksbank and their respective quality. The feedback sequences for the confusion with *Information Provision* and *Employee Attitude and Behavior* mention changes concerning their local

bank and that they are usually helped well. Since there have been more examples concerning those two topics on which the classifier had been trained on, the overlapping of the content within the topics has led to this confusion. Lastly, *Price and Quality* has been confused with *Quality and Offering*. The feedback sequence was the following:

- *The automatic interest rate adjustment after repaying the mortgage works fine.*

The statement mentions paying off their mortgage and the interest rate - both also things that can be mentioned in feedback labeled as *Price and Quality*. Since both topics are closely related, they have been merged in one of the data adaptation steps. The first topic that has been correctly identified is *Employee Contact*, two out of five possible times. The two sequences are:

- *Fast services and good accessibility.*
- *Great accessibility*

Because the content should be employee-related, it was expected that most confusion would be made with *Employee Knowledge & Skills* and *Employee Attitude and Behavior*. Instead, the remaining statements were confused with:

- *The SNS has an accessible system for extra repayments* – **Digital Options**
- *Fast response and great advice.* – **Information Provision**
- *Good service and staff is easily accessible.* – **Employee Knowledge & Skills**

Almost all sequences included the word "accessible", however, it is not only specifically found in the topic *Employee Contact* but also in the majority of the other topics, thus leading to this confusion.
*Digital Options* has been correctly predicted six out of eleven times. These feedback statements consisted of the following:

- *Easy app and website will information.*
- *Clear prognosis of the shares visible online.*
- *Very satisfied with your service. The app is easy to use.*
- *Website is simple but kind of old.*
- *Great app, I like that the digipass is not required anymore.*
- *Very simple to use website.*

All statements mention either the "website", the "app", or "online options". However, even though the following statements share the same vocabulary, the classifier was not able to classify them as *Digital Options*.

- *My advisor is always accessible via WhatsApp. He arranged a mortgage in a few weeks with super sharp interest offerings. More than satisfied!* – **Handling**
- *For a quick handling of the request a short form on the website is enough instead of direct phone contact.* – **Handling**

- *The app works great. All my questions are answered right away.* – **Employee Knowledge & Skills**
- *Digitally it is very easy to change the release of the mortgage.* – **Processes**
- *Why do I have to upload unnecessary info into the home folder? I already sent you my passport.* – **Processes**

Despite the usage of the related vocabulary, it also becomes apparent that the feedback sequences are way longer compared to the correctly predicted ones. The average for the short sentences is around eight words, whereas the longer sequences consist of an average of almost seventeen, which is more than twice as much as the short ones. Since the classifier does not take the content into account but only the individual words, this might have led to the confusion. Additionally, the feedback sequences, based on a human-based judgment, could also be labeled as other topics despite *Digital Options* - in this case, assigning multiple labels to one topic might come in handy.
*Price and Quality* has also been correctly predicted six times; however, the possible total could have been 19. It was confused with *Digital Options, Employee Knowledge & Skills, Handling, Information Provision,* and *Overall Experience.* The correctly predicted sequences were:

- *I have the opportunity to adjust my interest rates later.*
- *Drop the fine on your interest rates.*
- *In terms of interest height etc. everything is fine.*
- *Durable, great service, and affordable interest.*
- *The offered mortgage interest improved.*
- *Low interest rates.*

All the sentences mention the "interest" or "interest rate" on some level, indicating that this might be a frequently used keyword the classifier identified to predict a sentence as *Price and Quality.* This assumption is underlined with the wrong predictions the classifier has made:

- *I do not want to receive emails about surveys.* – **Handling**
- *I am satisfied, but I pay more interest than needed.* – **Overall Experience**
- *Just gone according to plan* – **Overall Experience**
- *I chose the bank 12 years ago and extended 2.5 years ago. We could talk about lowering my interest rate.* – **Digital Options**
- *Everything is transparent. They check if an interest adjustment might be useful.* – **Information Provision**
- *Great quote.* – **Information Provision**
- *Very satisfied with the explanation and possibilities that are available.* – **Information Provision**

- *Always answer within a reasonable period of time* – **Employee Knowledge & Skills**

The feedback statements barely mention *interest*. However, after examining some statements it raises the question how trustworthy the gold annotations actually are. For instance, the first sentence in the list above mentions that the person no longer wishes to receive emails about surveys and instead of labeling it as *No topic found* or another more fitting topic it has been labeled as *Price and Quality*.
SVM was able to correctly predict *No topic Found* 13 out of 32 times. Correctly predicted sentences were feedback statements such as:

- *Do a realistic proposal.*
- *well deserved*
- *I have not experienced anything negative*
- *You are great :)*
- *Everything went well*

Since punctuation was removed in the pre-processing step, the ":)" did not affect the classification process. However, the statements were also confused with *Employee Knowledge & Skills, Handling, Information Provision, Overall Experience, Price and Quality, Employee Contact,* and *Processes*. Especially the confusion with *Overall Experience* is understandable; several sequences that appear above appear in similar constellations, labeled as *Overall Experience*, within the train dataset. The most confusion, however, was around *Information Provision* with sequences like:

- *They just offered a mortgage.*
- *They did what I asked for*
- *Good contact*
- *Can you call when I have an appointment?*
- *well guided*

After reading these sequences, it is quite unclear as to why they have been confused with *Information Provision*. However, after looking through the train data, many sequences do resemble the ones above and thus create the confusion leading again to the question of how well the gold annotations represent the content.
*Employee Knowledge & Skills* was correctly detected 20 out of 43 times and mostly confused with *Employee Attitude and Behavior* and *Information Provision* but also with *Handling* and *No topic found*. Correctly predicted sequences mentioned how well the employees were informed and knowledgeable as well as communicative and well prepared. The most confusion, however, appeared with *Employee Attitude and Behavior* which is reasonable as both topics focus on employee-related feedback and can thus overlap greatly. Besides, it was also frequently confused with *Information Provision* concerning statements like:

- *They do what I ask for.*

- *Everything went smoothly and great and I trust my advisor.*
- *Good service and good information.*
- *Correct treatment and great advice*

At first glance, the sentences do not have something in common and do not perfectly fit neither into *Employee Knowledge & Skills* nor *Information Provision*. However, it can be assumed that since the most variety appears in the training samples of *Information Provision*, it leads to the wrong assumption that many phrases belong to this topic.
*Processes*, however, was confused with a wide array of other topics. Besides the 31 correctly classified feedback statements, the statements were confused with *Digital Options, Employee Attitude and Behavior, Handling, Information Provision, No topic found, Overall Experience,* and *Price and Quality*. Especially the confusion with *Handling* and *Overall Experience* comes with no surprise, as they often contain feedback statements like:

- *Everything goes well*
- *Everything went well*
- *Nothing to complain*

This again raises the question if the feedback statements should have been annotated differently, as the overlapping of several feedback statements only leads to confusion for the classifiers. Mainly because the same confusions appear with feedback statements that should have been either *Handling* or *Overall Experience* and are misclassified as one of them or *Processes*.
As previously examined, *Employee Attitude and Behavior* and *Information Provision* have led to a lot of confusion for most topics. These two are the most represented ones in the dataset and thus provide the greatest variety regarding their feedback statements. Since the statements are not always perfectly annotated or could belong to several topics simultaneously, the classifier often confuses the other topics with those two.
Interestingly *Employee Attitude and Behavior*, correctly predicted 92 out of 113 times, was also confused with *Information Provision* several times:

- *Nice employee - clear and knowledgeable. I indicated that I might need to redeem something at some point.*
- *Very satisfied with the personal contact.*
- *They also help well on the phone and explain everything well*
- *Sympathetic advisor on the phone*

Even though the sentences do mention the behavior and attitude of the employees, SVM is not able to detect that and misclassifies it as the way information is being provided.
Lastly, *Information Provision* was correctly predicted 175 out of 202 times. However, the remaining feedback statements were confused with all other ten topics. This again indicates much correlation in the content of *Information Provision* with the content of the other topics.

**Concluding Remarks: Topic Classification on the Original Dataset**

Finally, it can be said that the Support Vector Machine with the BoW TF-IDF approach correctly predicted more than half, 427 out of 607, of the topics. Most of the confusion arose around *Information Provision* and *Employee Attitude and Behavior*, which are the most represented classes within the dataset and thus provided the most variety but also overlap content-wise a lot with the other topics. Apart from that, the inconsistent gold annotations affect the performance negatively and cause a lot of confusion, for SVM and all other classification models as well. The classification worked especially well when feedback statements of certain topics shared the same vocabulary.
The outperformance of SVM compared to the transformer-based models comes as a surprise as they are the current state-of-the-art for classification tasks and are usually superior to traditional machine learning algorithms. However, this performance shift might be due to the nature of the feedback statements provided in this dataset; a lot of sentences are relatively short, only containing keywords, and are incomplete. BoW Tf-IDF is a keyword-based method that captures the important keywords more easily and does not take semantic meaning, the context of the words, or word order into consideration - whereas transformers are context-based. Additionally, the SVM has been specifically trained on this dataset. In contrast, the transformers have only been fine-tuned on it and have been pre-trained on massive datasets beforehand. The dataset with a total size of 6,071 and a training size of 1,568 is comparatively small. It does not seem sufficient for fine-tuning a transformer and for the model to learn the necessary information to interpret them. Additionally, the weak support of test samples in some topics appears to be too few for the classification algorithms to evaluate the performance.
To evaluate if the performance of the TC can be improved by increasing the data, we implemented data augmentation techniques which are being analyzed in the following section.

### 4.2.2 Analysis: Topic Classification on Augmented Datasets

The data augmentation methods focused on back-translating 10% and 20% per topic from English to Dutch and back to English. As already observed above, this technique generally did not improve the performance. The following sections focus on a deeper analysis of the results to examine the causes behind the decline.
First, the 10% back-translated dataset still performs best with the SVM BoW TF-IDF approach with an F1-score of 0.527. However, the performance declined from 0.537 to 0.527, i.e., instead of correctly predicting 427, or more, out of 607, 423 were predicted. The confusion matrix in Figure 4.2 provides an overview of the correct and wrong predictions.

Concerning *Quality and Offering*, *Employee Contact*, and *Digital Options* no changes were detected in the predictions. This indicates that the back-translation did not support classifying underrepresented topics. However, *Price and Quality* lost one true positive and instead gained misclassifications concerning *Handling*. One was previously wrongly classified as *Overall Experience*, and the other one was actually correctly classified when trained on the original dataset. *No topic found* gained two true positives, rising from 13 to 15. Some changes can be detected when comparing the wrongly predicted statements; *Employee Attitude and Knowledge* and *Overall Experience* each


Figure 4.2: Confusion Matrix SVM BoW TF-IDF Back-translation 10% Dataset

lose a misclassification. *Employee Knowledge & Skills* and *Handling*, however, gain one each. *Information Provision* loses two misclassifications. Some of the changes were detected in the following sentences:

- *Did what I asked.* – misclassified as **Employee Knowledge & Skills**, formerly misclassified as **Information Provision**
- *Oh well, be less like HEMA.* – misclassified as **Handling**, formerly misclassified as **Employee Attitude and Behavior**
- *Nothing really new to me* – misclassified as **No topic found**, formerly misclassified as **Overall Experience**

This topic is the one with the most changes due to the back-translation. Since *No topic found* carries a lot of variety and ambiguity, it is understandable that with the random data augmentation, the noise of this topic also increases. *Employee Knowledge & Skills* gains one true positive and loses the misclassification with *Information Provision* concerning the statement "Our advisor understands his profession". *Processes* loses four true positives which are added to *Employee Attitude and Behavior*, *Handling* and *Overall Experience*. Since the classifier also had issues with these labels when trained on the original dataset, it is no surprise that the misclassifications were made after the data augmentation. The true positives for *Overall Experience* remain the same; however, one statement, "Easily approachable." moves from the misclassification about *Processes* to *Employee Contact*. This is interesting since *Employee Contact* belongs to the underrepresented topics and rarely causes any confusion - indicating that the back-translation helped the classifier to at least recognize the topic. *Employee Attitude and Behavior* loses two true positives, which are added to *Overall Experience* and *Processes* - this again comes to no big surprise as there have been misclassification issues concerning those topics beforehand. The same can be observed for *Information Provision* which loses one true positive that is added to *Employee Knowledge & Skills*.

Overall, the 10% back-translation did not cause many changes and somewhat impaired the performance. As a next step, 20% of the data was being back-translated in order to evaluate if an amount of this augmentation affects the performance positively. It caused the best-performing model, SVM BoW TF-IDF, to drop its performance from 0.537 trained on the original to 0.527 trained on the 10% back-translated dataset and now to 0.520 - being outperformed by BERT with an F1-score of 0.525. Figure 4.3 shows the number of true positives and misclassifications, which will be analyzed shortly in the following sections. In total, 417 out of 607 feedback sequences were correctly identified, which is a drop of ten compared to the original dataset and a drop of six compared to the other back-translated dataset.

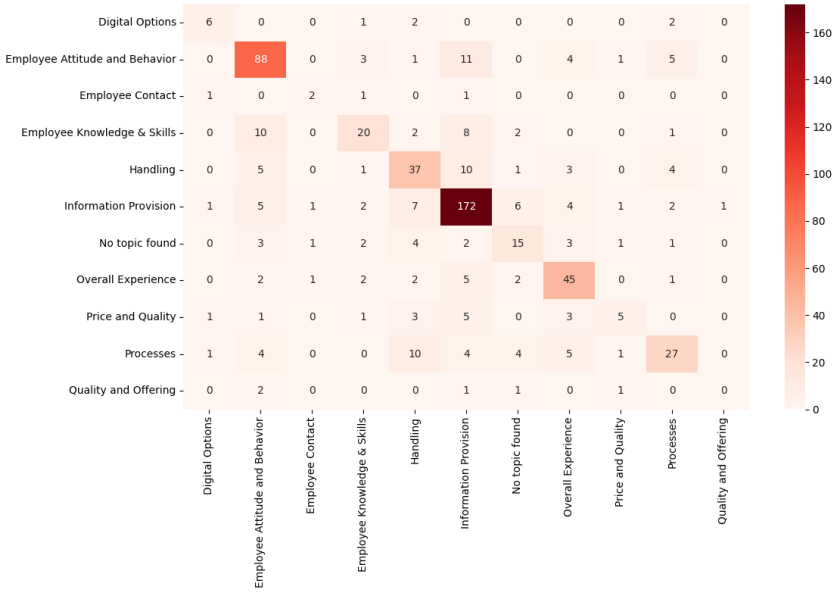



Figure 4.3: Confusion Matrix SVM BoW TF-IDF Back-translation 20%

There are still some changes that can be observed when comparing the performances of the three datasets. However, compared to the other classifiers, this performance was not superior, so this analysis will be relatively short and only focus on the main similarities and differences.
Once again, the underrepresented topics, *Quality and Offering, Employee Contact*, and *Digital Options* were not affected by the augmentation. There is no difference between the original dataset and the 10% back-translated one. It indicates that the added variety did not help sufficiently for the classifier to grasp the content of those topics. Additionally, many topics also overlap in their number of true positives with those from the 10% back-translated one; this is, for instance, the case for *Price and Quality, No topic found, Processes,* and *Handling*. For no topic, the true positives increased compared to the 10% back-translation. Besides that, the biggest differences compared to the original dataset happened with *Processes* losing four true positives, *Employee Attitude and Behavior* losing four, and *Information Provision* losing three. *Information Provision* and *Employee Attitude and Behavior* are still the causes of the majority of the confusion.

SVM was outperformed by several models, BERT and RoBERTa. BERT provided an F1-score of 0.525, the first time in this project that a transformer-based model outperforms the traditional machine learning algorithm SVM with BoW TF-IDF. However, this F1-score does still not reach the one SVM provided when trained on the original dataset, 0.537. When fine-tuned on the original dataset, BERT provided an F1 score of 0.517 and outperformed four other classification algorithms. In total, BERT could correctly predict 442 out of the 607 feedback statements. The confusion matrix in Figure 4.4 gives an insight into the performance after BERT has been fine-tuned with the 20% back-translated dataset.

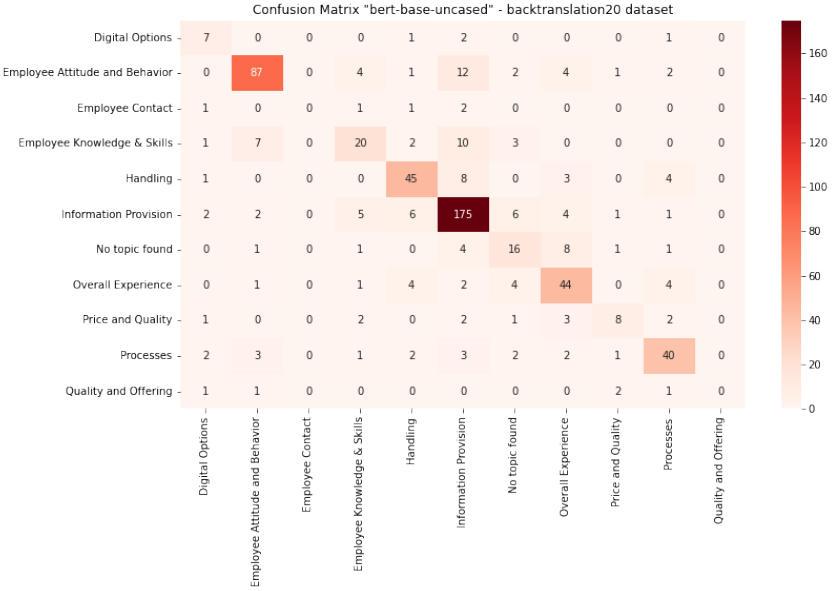



Figure 4.4: Confusion Matrix BERT Back-translation 20%

Since the performance is still lower than SVM's on the original dataset, the results are not being compared; some results will be compared to the performance of BERT after being fine-tuned on the original dataset to see the improvements and setbacks. Nevertheless, the following section dives into the true positives and misclassifications that BERT could predict after being fine-tuned on a 20% back-translated dataset.

BERT was not able to catch the feedback statements for *Quality and Offering* nor the ones for *Employee Contact*. Instead, the statements were confused with several other topics. Concerning *Quality and Offering* the classifier predicted the topics *Employee Attitude and Behavior, Digital Options, Price and Quality,* and *Processes*. The most confusion appeared with *Price and Quality* and the following statements:

- *Give the branch manager some more freedom to serve existing clients. That way they do not have to switch to another lender and leaves satisfied clients.* – formerly misclassified as **Processes**
- *Automatic interest rate adjustment after repayment on mortgage would be nice* – formerly misclassified as **Price and Quality**

The first statement indicates that the employees should have more freedom concerning the offers they propose, and the second statement mentions the interest rate and

repayment of a mortgage - both statements overlap with the content of *Price and Quality* and thus making the confusion reasonable. *Employee Contact* was confused with *Digital Options, Employee Knowledge & Skills, Handling,* and *Information Provision.* These issues can also be found when looking at BERT's predictions when fine-tuned on the original dataset - except the statement "Good accessibility" which was formerly misclassified as *Digital Options* and is now misclassified as *Information Provision.* It shows that the back-translated statements that were added to the dataset were not able to clear up the confusion for the two topics analyzed so far. However, the back-translation helped with the statement "I can handle requests with a short form on the website and when I have more questions, direct (telephone) contact with my advisor.". It used to be misclassified as *Information Provision* but was now added to the true positives. Concerning *Price and Quality* not many differences can be seen when comparing the performances. Eight out of 19 were correctly predicted, and the remaining eleven mostly overlap in their misclassification with the original dataset. The only differences were noticed concerning the following statements:

- *Very happy with the explanation and the options that are available for us to make it as economical as possible for us* – misclassified as **Processes**, formerly misclassified as **Information Provision**
- *Well, I have been a customer for 12 years. Maybe we can talk about the height of the interest* – misclassified as **Processes**, formerly correctly classified as **Price and Quality**
- *stick to the agreements regarding the interest discount due to salary on checking account or enough transactions. After agreement, we had to wait an extra month the offer was too high. We can't buy a house like that.* – correctly predicted as **Price and Quality**, formerly misclassified as **Information Provision**
- *Overall satisfied, but we pay more interest than we need to* – misclassified as **Overall Experience**, formerly correctly classified as **Price and Quality**

It can be seen that there is more confusion with *Processes* but less with *Information Provision* - the back-translation might have helped in the understanding for the latter but now created more confusion for the former. This is also the case for the last sentence, and it can be assumed that because of the terminology *satisfied*, the confusion was created as a majority of the content of *Overall Experience* includes this word.
*No topic found* has been correctly identified 50% of the time. Besides, it has been confused with a wide variety of other topics which is due to its noisy and unstructured content; *Employee Attitude and Behavior, Employee Knowledge & Skills, Information Provision, Overall Experience Price and Quality,* and *Processes.* The only difference is one sequence that was formerly correctly predicted and is now misclassified as *Information Provision* and the statement "I have nothing to get with you." is now correctly classified. Concerning *Employee Knowledge & Skills* it can be observed that the data augmentation indeed helped. The following sentences were misclassified when fine-tuned on the original dataset but are now correctly identified:

- *Sympathetic and skilled advisor.* – used to be misclassified as **Employee Attitude and Behavior**
- *Very professional advice. Very well prepared. Clear, and keeps the future in mind.* – used to be misclassified as **Information Provision**

- *Correct treatment and expert contact.* – used to be misclassified as **Information Provision**

Not only did the data augmentation help here with content-related topics, i.e., *Employee Attitude and Behavior*, but it also decreased the confusion with *Information Provision*. This can also be observed concerning *Processes*, statements that used to be confused as *Information Provision* but are now correctly predicted. However, the back-translation caused additional confusion and now misclassified statements that were predicted correctly beforehand. For instance, " Everything went well with my direct debit.", is now misclassified as *Overall Experience*. This can be due to the wording "everything went well", which, like the word "satisfied", frequently appears within the training samples. Many statements in the test set containing this kind of wording, however, were correctly identified as *Overall Experience*; for instance:

- *I am satisfied.*
- *Everything went correctly.*
- *We are satisfied.*
- *Very satisfied!*

Although BERT was able to identify 44 out of 60 statements correctly, there were still some issues with the predictions. Especially sentences like "Everything went well" were initially correctly predicted and after the back-translation are now misclassified as *Handling*. This indicates that in the back-translated dataset, more statements with this phrasing appear labeled as *Handling* instead of *Overall Experience*. When examining the results concerning *Handling*, this claim becomes apparent, as the feedback sequences "Everything went well" and "It went well" appear in the gold labels as *Handling* and are misclassified as *Overall Experience* - pointing out again the many issues within the rule-based gold annotations. Besides that, the decreasing confusion with *Information Provision* can also be observed concerning the true positives of *Handling*. Even though this decrease can not be observed concerning *Employee Attitude and Behavior*, a decrease in the confusion with *Employee Knowledge & Skills* can be observed. The number of true positives does not change for *Information Provision*; however, some minor changes and issues have been observed after extensively analyzing the predictions, primarily due to the inconsistent gold labels. A selection can be found here:

- *Nice and clear conversation where they took enough time for me.* – formerly correctly predicted, now misclassified as **Handling**
- *This mortgage is not mine. Could you please call me about this tomorrow so we can figure this out??* – formerly misclassified as **Overall Experience**, now **Information Provision**
- *I would prefer a video call because of Covid.* – formerly correctly predicted, now misclassified as **Digital Options**

As it can be seen in the examples above, the data augmentation caused some issues with the predictions. However, the issues can also be traced back to the annotations as it is not comprehensible why, for instance, the second statement should be classified as *Information Provision* after all. The misclassification, especially with the third one, is understandable as the statement mentions *Digital Options*. Thus either an updated gold label or a multi-labeling might be beneficial.

**Concluding Remarks: Topic Classification on Augmented Datasets**

Overall it can be concluded that the back-translation did not aid the performance of traditional machine learning classifiers - quite the contrary, it caused a decrease in their performance. However, it has been observed that the augmentation supported the performance for transformer-based models and caused an increase from 0.517, concerning BERT, to 0.523 and then 0.525. Admittedly, the performance is still lower than the one provided by SVM BoW TF-IDF when trained on the original dataset. However, if the chosen classification algorithm is a transformer-based model, then this data augmentation method might positively impact it. The shift in the performance is most likely due to the increase in data that helps BERT grasp the relations between topics and feedback statements. Additionally, since the data was machine-translated before, there might not be enough variety in the dataset concerning the vocabulary, which is helpful for a keyword-based approach like BoW TF-IDF. However, the back-translation might have changed that by creating more variety. The variety often times appears for verbs and adjectives, for instance the sentence in Figure 3.6 changes the verb *wanting* to *trying*. Because it adds more variety but does not change the overall context, BERT is fine-tuned on more content, thus increasing its performance. On the contrary, the back-translation might also contribute to decreasing some variety. For the terms *advisor* and *consultant*, the former appears 97 times in the original dataset, 117 in the 10% back-translated one, and 135 in the 20% back-translated train dataset. In contrast, the latter appears 35, 37, and 38 times. For a keyword-based classifier, the representation might not be enough to be able to correctly identify it, whereas BERT is able to do so. The back-translation can also include translation errors due to the nature of the original feedback statements. That can be seen, for instance, concerning the word *adviser* which was always back-translated to either *advisor* or *consultant* as the frequency in all three datasets remains the same. Since the augmentation method supported BERT's performance, one can still experiment with back-translation as an augmentation tool, for instance, by only back-translating the minority classes or back-translating more than 10 or 20% of the data. This, however, extends the scope of this research project. The performance of transformer-based models might also be improvable by changing the number of training epochs.
Additionally, it has been observed prevalently that the gold annotations do not follow consistent guidelines and thus affect the performance of all classification algorithms as the errors were implemented in the back-translation due to the randomized approach.

### 4.2.3 Analysis: Merging Topics

As seen in the analysis above, the number of topics and their overlapping content can contribute to confusion. To avoid this and create more balanced datasets, two merging approaches were chosen as additional experiments to quantify the extent to which the models and their performance can be improved. The experiments also serve to answer the research subquestion about the required minimum number of training samples in order to receive a reasonable performance. One approach merged the employee-related and finance-related topics into the topics called *Employee* and *Price and Quality* and the second one merges underrepresented topics into one topic called *Other*.

The performance of RoBERTa will be analyzed in the following section since, compared to the other traditional machine learning models and especially to the two

transformer-based models, it provided the best performance of a macro averaged F1-score of 0.668. In contrast, BERT and DistilBERT both have 0.661 and 0.637, respectively. RoBERTa is able to identify 458 out of the 607 feedback statements correctly. The merging caused the macro averaged F1-score to rise around 0.060 - from 0.528 to 0.668. The confusion matrix for RoBERTa's performance on the merged dataset can be seen in Figure 4.5. Since the statements that RoBERTa has been fine-tuned on are in the original and the merged dataset reasonably similar, and only the topics differ, many confused sentences are the same.

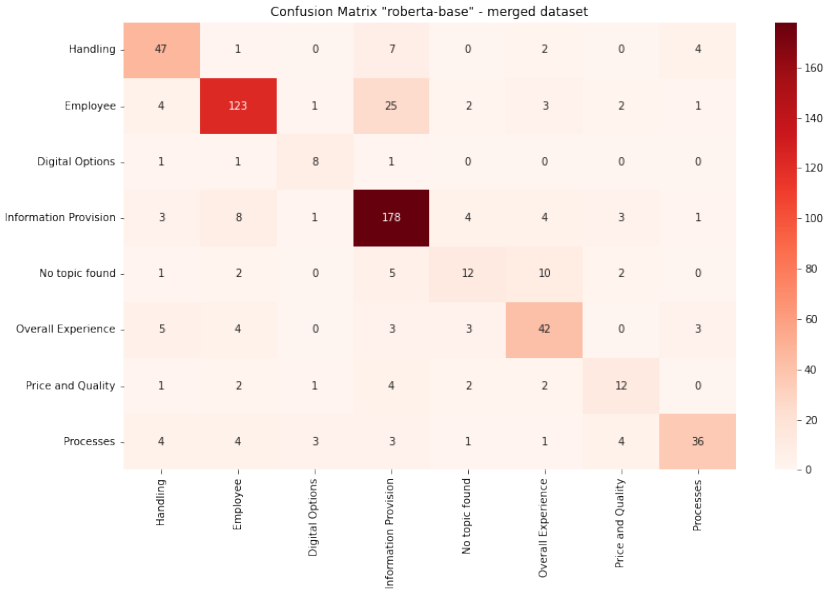



Figure 4.5: Confusion Matrix RoBERTa Merged Dataset

First and foremost, every topic was predicted correctly to some degree. Starting with *Digital Options* which was identified eight out of eleven times, compared to the original dataset using RoBERTa, it was nine out of eleven. The three remaining feedback sequences were confused with *Handling, Employee,* and *Information Provision.* The sentence that was confused with *Handling* is again the one mentioning "WhatsApp" and the work of the customer's advisor. The confusion with *Information Provision* has also occurred in the original dataset. *Employee* as a new topic caused a misclassification with the following sentence:

- *The app works great and has a nice interface. All questions are being answered right away.*

Even though "app" is mentioned as a digital option, RoBERTa is not able to detect it as such. It can be assumed that the second sentence caused the confusion - a multi-labeling would be beneficial here, annotating the statement as *Digital Option* and *Employee.* The next topic is *Price and Quality* which has been merged with *Quality and Offering*; thus, the total amount of possible true positives rises from 19 to 24. The model scores 12 out of 24, and the statements previously confused to be *Price and Quality* whereas they should have been *Quality and Offering* are now correctly identified. Most issues arose with *Information Provision*, but they are not due to the merging process as they appeared before. There are also two confusions with the new topic, *Employee*, concerning feedback about the help they received from employees and the service that has

been provided - also not due to the merging as it can be observed after being fine-tuned on the original dataset. Another sequence from *Quality and Offering* was misclassified as *Digital Options*; the same happened on the original dataset and can be traced back to the mentioning of the online service within the feedback. Apart from that, feedback statements that should have been classified as *Price and Quality* were classified as *No topic found* and *Overall Experience*.
The performance of the detection of the topic *No topic found* remains the same, meaning 12 out of 32 were correctly classified. These misclassifications can be traced back to the confusion with *Handling, Employee, Price and Quality*, but most of all with *Overall Experience* and *Information Provision*. In many cases, it can be observed that the misclassifications match the ones that RoBERTa made after being fine-tuned on the original dataset. Interestingly, the merging approach also affected the misclassifications with *Price and Quality* and *Employee*. The topic *Price and Quality* was wrongly predicted on a sentence that has previously been correctly classified. The sentence contains the word "surplus" and thus might be why it has been predicted as *Price and Quality*. The same issue appeared with a sentence misclassified as *Employee* - in the original dataset; it has been predicted correctly.
*Processes* has been classified 36 out of 56 times, four less compared to RoBERTa fine-tuned on the original dataset. The remaining 20 feedback statements are misclassified as *Handling, Employee, Digital Options, Information Provision, No topic found, Overall Experience* and *Price and Quality*. Due to a large number of confused topics, the overall distribution of the confusion is equally low. In many cases, the wrong predictions overlap with those of the RoBERTa fine-tuned on the original one. However, concerning *Handling* the model made more mistakes and misclassified sentences that had previously been correctly classified:

- *Everything is well arranged.*
- *Handy, fast, and simple delivery.*
- *The application was fast and smooth.*
- *The mortgage was quickly arranged with no problems.*

Something similar can be observed when looking at the sentences wrongly classified as *Digital Options*: all of them were correctly predicted in the original dataset — indicating that the merging might have influenced the understanding of the model concerning *Digital Options, Handling,* and *Processes*. Feedback statements that have been previously misclassified with an employee-related topic have been misclassified with *Employee* as well.
The performance of correctly predicting *Overall Experience* also suffered due to the merging, scoring 42 out of 60, which is four less than RoBERTa fine-tuned on the original dataset. While observing the predictions, the annotation issue mentioned beforehand became evident again. The feedback "Everything went well" appears as such and in closely related combinations several times within the dataset and is sometimes annotated as *Overall Experience* and sometimes as *Handling*. Something comparable has also been noticed for *Employee* as the phrase "Easily approachable" can be found gold labeled as an employee-related topic but also in *Overall Experience*.
For *Handling* the number of true positives rose from 41 to 47 out of 61. Confusion can be found concerning *Employee, Information Provision, Overall Experience,* and

*Processes*. Most confusion is due to *Information Provision*. However, all the confused statements appeared with the original dataset as well. While observing them, it has been noticed that one feedback states this:

- *I am satisfied but I did not want to participate in this study which also guarantees anonymity. Why am I receiving this then again?*

According to the gold labels, it should be *Handling*. However, it has nothing to do with the actual handling of mortgages but instead the handling of sending out surveys, so it raises the question again of how accurate and reliable the gold annotations are. Apart from this, *Handling* has been confused with *Employee, Overall Experience*, and *Processes* - compared to the classification on the original dataset, the misclassified sequences have not changed.
The second merged topic is *Employee*, which has been correctly predicted 123 out of 161 times. However, there was still some confusion, especially with *Information Provision* - RoBERTa fine-tuned on the original dataset misclassified only 18 feedback sequences as *Information Provision*. After merging, it is now up to 25. The other topics the sequences were confused with are *Handling, Digital Options, No topic found, Overall Experience, Price and Quality,* and *Processes*, but compared to *Information Provision* to a reasonably low distribution. The wrongly predicted sequences mostly contained the words "conversation", "advice", "contact", "question", "informed", "informative", and "helpful" that, besides describing the actions and skills of an employee, also can describe how the information was provided.
Lastly, *Information Provision* was caught 178 out of 202 times. The misclassified topics are *Handling, Employee, Digital Options, No topic found, Overall Experience, Price and Quality,* and *Processes* which is comparable to the performance of the original dataset. Interestingly, beforehand there has been no confusion with *Price and Quality*, but after merging, these sentences are being misclassified:

- *I got informed in an email that I can save on my mortgage. After a call I only reached someone that cannot deal with mortgages - that was weird.*

- *Everything follows my interest. The mortgage runs for 6 years now and the proposal for an extension came just in time.*

Both contain several headwords, *mortgage* and *interest*, that can also be an indication for *Price and Quality* instead of *Information Provision*, indicating that the classification might extract advantages from a multi-labeling approach.

To conclude, the merging process did help clarify some confusion for the model, especially concerning topics that overlap deeply in their content, i.e., *Employee* and *Price and Quality*. However, there still have been several issues, especially with *Information Provision* and *Overall Experience*. Since the frequency of the topics within the train dataset increased after the merging and the distribution can be claimed as a bit less unbalanced, the transformer-based models have classified every topic correctly to some extent. It was also demonstrated again that the gold annotations might not be as reliable as they should be and interfere with the classification task.

A different approach was to merge underrepresented topics, *Quality and Offering, Employee Contact, Price and Quality,* and *Digital Options*, into one called *Other*.

The performance will be evaluated in the following sections. Again, RoBERTa provided the best performance scoring a 0.670 macro average F1-score, whereas the other transformer-based models scored between 0.651 and 0.667. Overall, RoBERTa predicted 450 out of 607 feedback statements correctly. Figure 4.6 provides a first overview of the performance.

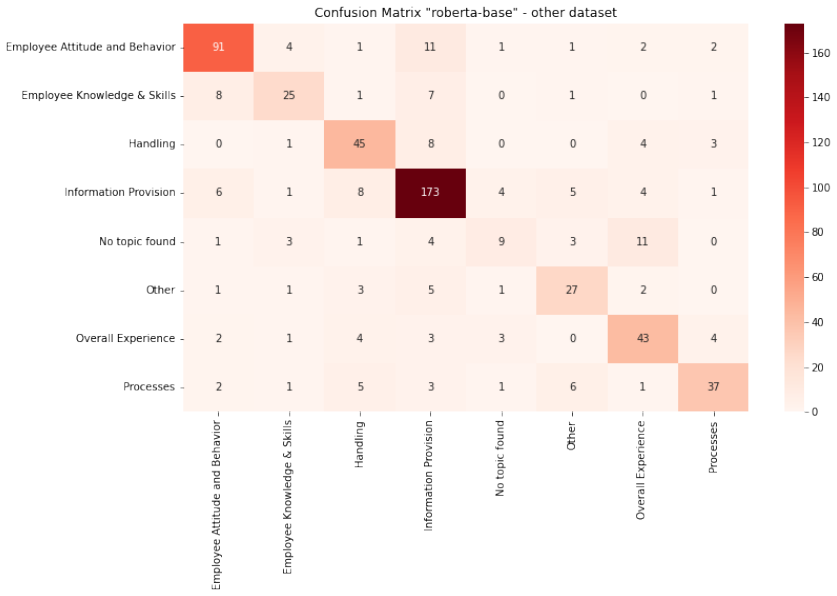



Figure 4.6: Confusion Matrix RoBERTa Other Dataset

*No topic found* was correctly identified 9 out of 32 times which is a decline compared to the performance when fine-tuned on the original dataset. Most of the confusion matches the one that RoBERTa had before. However, since this dataset works with different topics, some disparities have been observed. First and foremost, *Other* appeared in the confused sequences three times and consisted of the following sentences, which had previously been classified differently:

- *For me it was not profitable to join, maybe for someone else.* – misclassified as **Other**, formerly classified as **No topic found**
- *My monthly amount is now lower.* – misclassified as **Other**, formerly misclassified as **Price and Quality**
- *Be more risky.* – misclassified as **Other**, formerly classified as **No topic found**

One of the sentences had been misclassified as a topic of an underrepresented topic which is now part of *Other*. Thus it is comprehensible that it has also been misclassified with this dataset. The other two were initially predicted correctly, indicating that the new topic has confused the model due to its ambiguity. Even though *Other* made the impression so far that it contributed to the model's confusion, it has been identified 27 out of 40 times. However, it was confused with *Employee Attitude and Behavior, Employee Knowledge & Skills, Handling, Information Provision, No topic found,* and *Overall Experience*. Most issues arose with the feedback sequences that used to be *Price and Quality*; out of the 13 misclassified sequences, eight were part of this topic in the original dataset. There is no correlation concerning the wrongly

predicted topics, as these sequences appear in every one of them. That shows that the support of *Price and Quality* in the training data might be sufficient for the classifier to predict it. Merging it with the other underrepresented topics caused more confusion than clarification. Concerning the other former topics, most were correctly classified as *Other*, and the remaining four misclassifications have been made beforehand. However, this shows that even though the topic's content is quite ambiguous, the merging benefitted the three most underrepresented topics.
*Employee Knowledge & Skills* profited from this topic adaptation approach as the number of true positives rose from 20 to 25. Especially the confusion with *Information Provision* was affected by it and declined from ten to seven. There is one feedback sequence that had been confused with *Other*, which contains the word "Customizable" - prior to this, it had been confused with *Digital Options*. This example again shows that the gold annotations are not entirely reliable and thus can be scrutinized.
*Processes* did not profit from the topic adaptation as the score of correctly identified sequences went from 40 to 37 out of 56 possible ones. Most of the confused sequences align with the ones in the original experiment. However, *Handling* has been wrongly predicted five times concerning the following sentences:

- *It has been accomplished quickly and was not hard either.*
- *Very special that the offboarding process is greatly regulated.*
- *Fast and simple delivery.*
- *Our mortgage has been arranged without any problems and was very quick.*
- *The application was smooth and fast.*

Oddly, all these sentences were classified as *Processes* by RoBERTa when the model was fine-tuned on the original dataset. *Handling* was wrongly predicted only once with the sentence "Finalizing the mortgage via the advisor went smoothly." - which was correctly predicted using this adapted dataset. The overall confusion with *Handling* might indicate that the variety within *Other* could have led to the disarray. *Other* in itself has been wrongly predicted six times for sentences that were wrongly predicted as *Price and Quality* or *Digital Options* beforehand. The other misclassified topics mostly align with the ones from the original approach.
Concerning *Overall Experience* no major issues could have been determined after merging the underrepresented topics into one topic. The misclassified sequences mostly align with the ones of the original dataset and do not seem to be impacted by the new topic *Other*. The same applies to *Handling* and *Employee Attitude and Behavior*; *Other* did not affect the performance negatively but improved it by a few true positives.
Lastly, the topic adaptation affected the performance of *Information Provision* as the number of true positives dropped from 177 to 173. Most of the misclassifications match the ones from the original dataset. However, the confusion with *Employee Attitude and Behavior* doubled due to the following sentences:

- *We received a clear explanation about our possibilities. It was handled adequately and it was a personal approach.*
- *Great help with clear explanation. No obligations. Listened well to our input.*
- *Our advisor helped us well, took the time for us and explained everything clearly.*

The misclassification can be traced back to the fact that the employee's behavior is described as well and not only the way information is provided to the customer. Concerning *Other*, several misclassifications can be observed. Those were either correctly predicted by RoBERTa after being fine-tuned with the original dataset or belonged to one of the underrepresented topics:

- *Usually they send my things to my new address, but Wednesday I received the mail at my old address.* – **Correctly predicted**
- *The banking online goes well. It is reasonable easy.* – **Digital Options**
- *I would prefer a video call because of Covid.* – **Correctly predicted**
- *The amount last month was much more.* – **Price and Quality**

#### Concluding Remarks: Merging Topics

To draw a short interim conclusion, it can be said that merging underrepresented topics was partially helpful before describing the succeeding adaptation approaches. The merging caused an increase in the performance - which was expected as the number of topics was reduced.
The misclassifications indicated that the topic *Price and Quality* was sufficient in its frequency and content for the model to have a deeper understanding of the topic itself, i.e., the representation of around 80 train samples seems to be enough for transformer-based and machine learning algorithms to identify the topics. That is also, to some extent, the case for *Employee Contact*, with a train sample of 46, as SVM BoW TF-IDF trained on the original train dataset proved that the topic can be identified even with a low representation. It indicates that the traditional machine learning algorithms compared to transformer-based ones, are able to detect topics with a lower frequency. Besides that, this merging approach proved helpful for an unbalanced dataset. However, even though it helps to improve the performance by restructuring and decreasing the number of topics, it causes a loss of fine-grained information about the customer's experience as it is not clearly labeled anymore.

### 4.2.4 Analysis: Data Reduction

Looking at the F1-scores of the individual topics, it becomes apparent that they rapidly decline in correlation with the sample size of the topics in the train data. Please refer to the different Appendices for a better overview of the scores. Figure 4.7 represents the F1-scores of each topic after training an SVM BoW TF-IDF classifier on the original dataset. The bubbles represent the sample size of each topic, and the topics are sorted after their representative sample size, starting on the left with the most frequent one. To examine if the number of examples of the minority class is sufficient for the classification algorithms and to balance the dataset out, the data has been reduced to its minimum using undersampling. Every topic appears 23 times within the train data. The topics were randomly undersampled from the original train dataset.

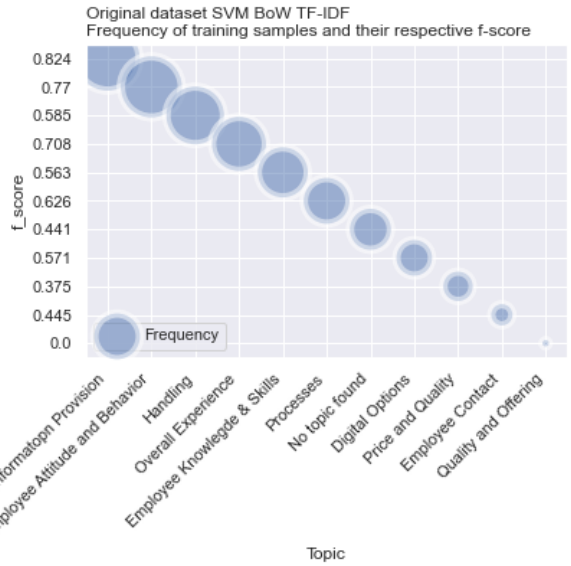



Figure 4.7: Individual F1-scores of the topics in relation to their sample size

As expected, the performance for all classification algorithms was quite low. However, Logistic Regression with BoW TF-IDF provided an F1-score of 0.342 - which is still a higher F1-score compared to other models trained on the original dataset, for instance, Naive Bayes with an F1-score of 0.250. The Logistic Regression was able to predict 250 out of the 607 feedback statements correctly. Figure 4.8 provides the confusion matrix.


Figure 4.8: Confusion Matrix Logistic Regression BoW TF-IDF undersampled train set

After a first glance at the matrix, it can be seen that instead of predicting the correct topic, Logistic Regression predicted *Information Provision* or *Employee Attitude and Behavior* - the two most represented ones in the test set. However, every topic was

caught at least once. Several reasons could have led to this highly unbalanced confusion. First, it might be because of the annotation issues that have been discovered before. Additionally, the misleading annotations were also incorporated since the undersampled feedback statements were randomly downsampled instead of manually picking the best-presenting ones. An analysis of the undersampled train dataset is required for a deeper understanding of the misclassification due to annotation issues. First and foremost, it has been noticed that many feedback statements fit into more than one topic. Sentences like

- *Fine conversation and clear explanation.*
- *Nice contact, my advisor is a very open person.*
- *Excellent customer contact.*

were all labeled as *Information Provision* and based on a human-judgment level it could also be one or several of the employee-related topics.
With the deletion of many feedback statements, much valuable information was lost that could have helped the classifier understand the relation of the feedback and its respective topic. Additionally, many of the feedback sequences the classifier has been trained on are quite short and do not provide sufficient context to label them as one of the eleven topics justifiably. Some examples of this are:

- *Great help*
- *Reachability*
- *Service*
- *good service*
- *good mortgage*

Since many feedback sequences are relatively short and incomplete, a Bag of Words TF-IDF approach has an advantage over transfer learning. This is because it does not consider the content and focuses on the frequency and inverse frequency of tokens. When looking through the predictions and comparing them to the best performing transformer-based model fine-tuned on the undersampled dataset, DistilBERT, it can be observed that many short sequences with similar vocabulary are correctly identified by Logistic Regression and almost always misclassified by DistilBERT. Some of these sentences are:

- *I have been helped well.*
- *Well helped.*
- *Perfectly helped*
- *helpful*

Instead of *Employee Attitude and Behavior*, DistilBERT either predicted *Employee Knowledge & Skills* or *Employee Contact*, but never the correct topic. This is not a topic-specific phenomenon and can be observed for all. However, this is only the case when fine-tuned on the undersampled dataset and does not occur when fine-tuned on the other larger datasets.

**Concluding Remarks: Data Reduction**

Lastly, it can be said that in this case, a support of 23 for each train sample is not sufficient enough for a transformer-based classifier to be fine-tuned on and predict sequences in a highly unbalanced test dataset but moderately enough to train the Logistic Regression BoW TF-IDF algorithm and provide somewhat correct predictions. This indicates that a number of 23 training examples per topic is still too little for a Logistic Regression to provide a reasonable performance concerning this dataset. However, it could be interesting to combine it with data augmentation approaches. One approach could be oversampling the underrepresented topics by a few. Another could focus on determining the average or the mean of the overall topic representation and then oversampling the topics below and undersampling the ones above. Instead of oversampling, one can also work with back-translation or similar linguistically focused augmentation approaches. This way, less information is lost and results in a more balanced dataset.

It has to be mentioned that the performance can change once the original training data is being undersampled again. The randomized approach does not distinguish between the sequences and does not pick the best-representing sequences per topic. Thus, after creating a new undersampled train set, the results might be worse or better; however, this extends the scope of this research project.

### 4.2.5 Analysis: Overall Concluding Remarks

In Section 4.1.5 the overall results of the classification algorithms and the different experimental setups were presented and shortly summarized. This section summarizes the main findings of the error analysis.
Surprisingly, SVM with a BOW TF-IDF feature representation outperformed the transfer learning models when training and fine-tuning them on the original dataset. As explained above, this might be due to the structure of the feedback statements, making it easier for a keyword-based method over a content-based one to identify the topics correctly. The TC on the original dataset already foreshadowed several issues that were intensified through the various data augmentation, reduction, and topic adaptation methods. One major issue consists of the inconsistent gold annotations, causing the majority of the misclassifications. The data augmentation in the form of back-translation did not help the traditional machine learning algorithms; however, it did improve BERT's performance. If this transformer-based model is the preferred classification technique, back-translating data batches could be beneficial for the performance. Furthermore, several other data augmentation techniques could be implemented and might help the transfer-learning- and machine-learning-based classifications.
The merging of the topics indicated that it is possible for this dataset to regroup the topics and that a smaller number might also be beneficial concerning the classifier's performance. Furthermore, the merging of the underrepresented topics showed that the topic *Price and Quality*, with a train sample of 81, caused the most confusion and is identifiable by several classifiers when trained on the original dataset. Hence, a representation of that amount is sufficient concerning the distribution of this dataset. Additionally, it was discovered that the number might be even lower for traditional machine learning. After training the SVM on the original train dataset, it became apparent that the topic *Employee Contact* with 46 train samples was identified several times correctly. Furthermore, after the data reduction, creating a balanced dataset

with a representation of 23 for each topic, the machine learning classifier was able to predict all topics to some extent. The minimum number of training samples is not fully generalizable concerning other datasets. Besides depending on the content of the statements in the train and test set, it also depends on the classification model that is being used. Consequently, it can be said that the transformer-based models require a larger amount of data, especially for minority classes, to be able to predict these topics.

# Chapter 5

# Conclusion and Discussion

## 5.1 Concluding Remarks about the Research

This research project focused on the supervised task of Automatic Topic Classification on customer feedback from the banking domain. The project aims to evaluate the performance of state-of-the-art Natural Language Processing models with regard to limited training data and an unbalanced distribution of a large number of classes. The execution of this project aims to improve Underlined's tools and to gain insights into the customer's experience and satisfaction which de Volksbank can use to improve its services.

The approach of this project contains selecting several classification methods; traditional machine learning and transfer learning. For the machine learning approach, Naive Bayes, Logistic Regression, and Support Vector Machine have been chosen and compared on a Bag of Words with TF-IDF approach as well as an embeddings approach using pretrained language models trained on financial data and data derived from news articles. For the transfer learning approach, the transformer-based models BERT, RoBERTa, and DistilBERT were compared, all having 4 training epochs, a batch size of 32, a maximum sentence length of 128, a learning rate of 3e-5, and AdamW Optimizer. SVM with a BoW TF-IDF feature representation proved its superiority on this classification task with a macro-averaged F1-score of 0.537.
Furthermore, due to the high unbalancedness of the dataset, several topic adaptation, data augmentation and reduction approaches were tested to improve the classifier's performance and gain an understanding of the minimum required training samples. Back-translation proved to be beneficial for BERT. Merging the topics and thus reducing the overall number improved the performance and indicated that for traditional machine learning algorithms, the number of required train samples is much lower compared to transfer learning ones.

### 5.1.1 Research Questions

The concluding remarks above summarized the answers to each research question. More detailed answers can be found below.

**Research Question:** Which classifier provides the best results for the Automatic Topic Classification task on customer feedback in the banking domain?

Concerning the provided dataset, a Support Vector Machine with a Bag of Words TF-IDF feature representation has proven itself to be the most efficient, with a macro-averaged F1-score of 0.537. It is somewhat unexpected as the transformer-based models are considered to be the state-of-the-art models, outperforming traditional machine learning. However, the transfer learning results were only slightly lower than the SVM; BERT scored 0.517, DistilBERT 0.525, and RoBERTa 0.528.
**Subquestion:** Will data augmentation techniques improve the classifiers' performance in the banking domain?

Back-translation was somewhat able to improve the performance of the transformer-based model BERT by increasing the number of representations per topic and creating more variety. The incorrect and inconsistent annotations contributed to the confusion of the models, as those were also included in the augmentation. The technique needs to be improved to affect the performances more positively - also for traditional machine learning classification algorithms - however, this extends the scope of this research project.

**Subquestion:** What is the minimum required number of training examples within each category for a classifier to provide a reasonable performance?

The project and especially the different topic adaptation methods in the form of merging topics with overlapping content and merging underrepresented topics into one topic indicated that a minimum of around 80 training examples is necessary to provide a reasonable performance for a transfer learning approach. The transformers did not predict topics that did not reach this amount of representation. 80 is not a fixed number and has only been observed for this particular dataset. Apart from that, both topic adaptation methods proved to be supportive of the classifier's performance.
The additional experiment implemented concerning randomly undersampling the data did not provide coherent results for the transfer learning approach. It showed that a minimum of 23 examples is only sufficient enough for the already overly represented majority class. However, after training a Logistic Regression with a Bag of Words TF-IDF, the algorithm was even able to predict the minority topics. Regardless, it cannot be concluded that 23 is the minimum required number of training examples per topic since the overall classification performance cannot be labeled as reasonable. Another observation that has been made concerning machine learning is that a frequency of around 40 train samples in the original train set was sufficient for SVM BoW TF-IDF to identify the topic. Overall, it can be seen that transfer learning models compared to traditional machine learning ones require a larger number of train samples. Further research might be necessary to determine a more exact required number for this classification task.

## 5.2 Future Work

Since this project was done within a strictly limited time frame, the classification task can be improved and adapted in many ways.
One major issue noticed while working on this project was the gold annotations. They were not adequately done and can be improved by changing it to human-based an-

notations instead of a rule-based approach. Proper annotation guidelines could help improve the quality of the gold labels. If the process shall remain automated, one can use the output of a Topic Modeling approach. Additionally, the feedback sequences can be annotated with several topics as one feedback can contain more than one, thus creating a multi-topic classification task. Another approach is to include subtopics, i.e., having fewer main topics, which can then be divided into many subtopics. For instance, the topics with employee-related content can be grouped into one main topic and then divided into several subtopics.
Since the data contains feedback that describes a customer's satisfaction, it is more than worthwhile to include a sentiment analysis after the topic classification to gain a deeper understanding of the customer's experience and sentiment about specific services. Apart from that, the data used in this project was originally in Dutch. It might be interesting to adapt the project to the Dutch data, using a transformer-based model pre-trained on Dutch data[1] or change the approach overall and turn it into a multilingual with a transformer-based model pre-trained on multilingual data[2] - so that data from different languages can be used in this classification task.
Furthermore, merging topics was proven to be an effective way to improve the classifier's performance. The improvement can be researched more thoroughly as there might be more topics with overlapping content that can be grouped. Besides the merging, data augmentation techniques have been included. However, especially linguistically speaking, there are many more, see Chaudhary (2020), that can be included and possibly be in favor of the model's performance. Those augmentation methods can be selectively implemented, focusing on the minority or majority classes or on datasets that have been adapted concerning their topics. Additionally, one can assign a higher weight to the underrepresented classes by modifying the classifier instead of the data. The dataset can be balanced out by calculating the average or median of the representations and oversample the topics below that score and undersample the ones above.
As shown in Chapter 4, the embeddings did not provide an adequate performance. However, the cause might have been the embedding model used, and other pre-trained models, like the embeddings[3] derived from FinBERT[4] might improve the performance. That leads to further research recommendations, fine-tuning other transformer-based models, such as FinBERT, or tuning the hyperparameters until the optimum performance for this task is found. Even though the traditional machine learning techniques did not succeed in their performance, feature engineering, as described in Section 3.2.1 can be implemented and might then outperform the transformer-based models.
From a company perspective, it might be interesting for Underlined to not only look at the sentiment analysis, multi-topic classification, and multilingual approaches but also to connect the results to the other scores in the data, such as the Key Performance Indicator (KPI) and the Customer Satisfaction Score (CSat) and the overall journey of the customer. Combining those results can help gain a deeper insight into the company's performance over time.

[1] https://huggingface.co/GroNLP/bert-base-dutch-cased
[2] https://huggingface.co/bert-base-multilingual-uncased
[3] https://github.com/abhijeet3922/finbert_embedding
[4] https://huggingface.co/ProsusAI/finbert

## 5.3 Limitations

As indicated above, the project was performed within a strictly limited and short time frame. However, this is not the only limitation that restricts and influences this project. One issue was the strict privacy policy concerning the data, making it impossible to use cloud-based Jupyter Notebooks such as GoogleColab[5] that provide a limited free access to GPUs. Instead, CPUs had to be employed within a controlled environment. It ensured that the sensitive data was not being shared with third parties, letting transformer-based models run for several hours each and negatively contributing to the time frame issue mentioned beforehand.
Overall, one task-specific limitation is the necessity of an annotated dataset. In this case, it was created using a rule-based approach and contributed to many misclassifications since the annotations were not completely correct and inconsistent. Apart from that, there is a fixed number of topics, and the model cannot detect new ones. That can be achieved using an unsupervised Topic Modeling approach.
Another limitation concerning the data comes in the form of the translation. Through the machine translation, variety and spelling errors are lost, making it questionable if the trained and fine-tuned models can be classified as robust. Concerning the translation issue, back-translating the data creates ethical concerns and disadvantages. It is risky to count on synthetic data instead of natural data. The data could contain biases derived from the original data, has a lower quality, and create environmental costs while additionally being a time-consuming task.

[5] `https://colab.research.google.com/`

# Appendix A

# Results Traditional Machine Learning

## A.1 Bag of Words and TF-IDF

### A.1.1 Original Dataset

| Topic | Precision | Recall | F-Score | Support |
|---|---|---|---|---|
| Digital Options | 1.000 | 0.000 | 0.000 | 11 |
| | 0.833 | 0.455 | 0.588 | 11 |
| | 0.600 | 0.545 | 0.571 | 11 |
| Employee Attitude and Behavior | 0.651 | 0.628 | 0.640 | 113 |
| | 0.714 | 0.796 | 0.753 | 113 |
| | 0.730 | 0.814 | 0.770 | 113 |
| Employee Contact | 1.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| | 0.500 | 0.400 | 0.445 | 5 |
| Employee Knowledge & Skills | 1.000 | 0.023 | 0.045 | 43 |
| | 0.630 | 0.395 | 0.486 | 43 |
| | 0.714 | 0.465 | 0.563 | 43 |
| Handling | 0.758 | 0.410 | 0.532 | 61 |
| | 0.596 | 0.557 | 0.576 | 61 |
| | 0.581 | 0.590 | 0.585 | 61 |
| Information Provision | 0.466 | 0.975 | 0.630 | 202 |
| | 0.669 | 0.911 | 0.771 | 202 |
| | 0.785 | 0.866 | 0.824 | 202 |
| No topic found | 1.000 | 0.031 | 0.061 | 32 |
| | 0.556 | 0.156 | 0.244 | 32 |
| | 0.481 | 0.406 | 0.441 | 32 |
| Overall Experience | 0.889 | 0.400 | 0.552 | 60 |
| | 0.677 | 0.700 | 0.689 | 60 |
| | 0.657 | 0.767 | 0.708 | 60 |
| Price and Quality | 1.000 | 0.000 | 0.000 | 19 |
| | 0.667 | 0.211 | 0.320 | 19 |
| | 0.462 | 0.316 | 0.375 | 19 |
| Processes | 0.769 | 0.179 | 0.290 | 56 |
| | 0.711 | 0.482 | 0.574 | 56 |
| | 0.721 | 0.554 | 0.626 | 56 |
| Quality and Offering | 1.000 | 0.000 | 0.000 | 5 |
| | 1.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| accuracy | | | 0.542 | 607 |
| | | | 0.672 | 607 |
| | | | 0.703 | 607 |
| macro avg | 0.867 | 0.241 | 0.250 | 607 |
| | 0.641 | 0.424 | 0.455 | 607 |
| | 0.566 | 0.520 | 0.537 | 607 |
| weighted avg | 0.701 | 0.542 | 0.470 | 607 |
| | 0.666 | 0.672 | 0.644 | 607 |
| | 0.692 | 0.703 | 0.693 | 607 |

Table A.1: Overview results traditional machine learning on the original dataset, within a cell from top to bottom: Naive Bayes, Logistic Regression, and Support Vector machine

### A.1.2 Merged Dataset

| Topic | Precision | Recall | F-Score | Support |
|---|---|---|---|---|
| Digital Options | 1.000 | 0.000 | 0.000 | 11 |
| | 0.833 | 0.455 | 0.588 | 11 |
| | 0.600 | 0.545 | 0.571 | 11 |
| Employee | 0.657 | 0.807 | 0.724 | 161 |
| | 0.702 | 0.832 | 0.761 | 161 |
| | 0.773 | 0.807 | 0.790 | 161 |
| Handling | 0.742 | 0.377 | 0.500 | 61 |
| | 0.607 | 0.557 | 0.581 | 61 |
| | 0.587 | 0.607 | 0.597 | 61 |
| Information Provision | 0.541 | 0.906 | 0.678 | 202 |
| | 0.734 | 0.886 | 0.803 | 202 |
| | 0.789 | 0.851 | 0.819 | 202 |
| No topic found | 1.000 | 0.031 | 0.061 | 32 |
| | 0.571 | 0.125 | 0.205 | 32 |
| | 0.458 | 0.344 | 0.393 | 32 |
| Overall Experience | 0.885 | 0.383 | 0.535 | 60 |
| | 0.707 | 0.683 | 0.695 | 60 |
| | 0.638 | 0.733 | 0.682 | 60 |
| Price and Quality | 1.000 | 0.000 | 0.000 | 19 |
| | 0.667 | 0.250 | 0.364 | 19 |
| | 0.538 | 0.292 | 0.378 | 19 |
| Processes | 0.769 | 0.179 | 0.290 | 56 |
| | 0.750 | 0.482 | 0.587 | 56 |
| | 0.714 | 0.536 | 0.612 | 56 |
| accuracy | | | 0.610 | 607 |
| | | | 0.708 | 607 |
| | | | 0.720 | 607 |
| macro avg | 0.824 | 0.335 | 0.348 | 607 |
| | 0.696 | 0.534 | 0.573 | 607 |
| | 0.637 | 0.589 | 0.605 | 607 |
| weighted avg | 0.689 | 0.610 | 0.551 | 607 |
| | 0.702 | 0.708 | 0.686 | 607 |
| | 0.712 | 0.720 | 0.712 | 607 |

Table A.2: Overview results traditional machine learning on the merged dataset, within a cell from top to bottom: Naive Bayes, Logistic Regression, and Support Vector machine

### A.1.3 Merged "Other" Dataset

| Topic | Precision | Recall | F-Score | Support |
|---|---|---|---|---|
| Employee Attitude and Behavior | 0.651 | 0.628 | 0.64 | 113 |
| | 0.714 | 0.796 | 0.753 | 113 |
| | 0.732 | 0.796 | 0.763 | 113 |
| Employee Knowledge & Skills | 1.000 | 0.023 | 0.045 | 43 |
| | 0.630 | 0.395 | 0.486 | 43 |
| | 0.714 | 0.465 | 0.563 | 43 |
| Handling | 0.758 | 0.410 | 0.532 | 61 |
| | 0.603 | 0.574 | 0.588 | 61 |
| | 0.581 | 0.590 | 0.585 | 61 |
| Information Provision | 0.466 | 0.975 | 0.630 | 202 |
| | 0.698 | 0.906 | 0.789 | 202 |
| | 0.788 | 0.866 | 0.825 | 202 |
| No topic found | 1.000 | 0.031 | 0.061 | 32 |
| | 0.556 | 0.156 | 0.244 | 32 |
| | 0.464 | 0.406 | 0.433 | 32 |
| Overall Experience | 0.889 | 0.400 | 0.552 | 60 |
| | 0.689 | 0.700 | 0.694 | 60 |
| | 0.657 | 0.767 | 0.708 | 60 |
| Processes | 0.769 | 0.179 | 0.290 | 56 |
| | 0.722 | 0.464 | 0.565 | 56 |
| | 0.732 | 0.536 | 0.619 | 56 |
| Other | 1.000 | 0.000 | 0.000 | 40 |
| | 0.607 | 0.425 | 0.500 | 40 |
| | 0.576 | 0.475 | 0.521 | 40 |
| accuracy | | | 0.542 | 607 |
| | | | 0.684 | 607 |
| | | | 0.707 | 607 |
| macro avg | 0.817 | 0.331 | 0.344 | 607 |
| | 0.652 | 0.552 | 0.577 | 607 |
| | 0.655 | 0.613 | 0.627 | 607 |
| weighted avg | 0.701 | 0.542 | 0.470 | 607 |
| | 0.675 | 0.684 | 0.663 | 607 |
| | 0.702 | 0.707 | 0.700 | 607 |

Table A.3: Overview results traditional machine learning on the other dataset, within a cell from top to bottom: Naive Bayes, Logistic Regression, and Support Vector machine

### A.1.4 Back-translation 10%

| Topic | Precision | Recall | F-Score | Support |
|---|---|---|---|---|
| Digital Options | 1.000 | 0.000 | 0.000 | 11 |
| | 0.857 | 0.545 | 0.667 | 11 |
| | 0.600 | 0.545 | 0.571 | 11 |
| Employee Attitude and Behavior | 0.636 | 0.664 | 0.649 | 113 |
| | 0.698 | 0.796 | 0.744 | 113 |
| | 0.732 | 0.796 | 0.763 | 113 |
| Employee Contact | 1.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| | 0.400 | 0.400 | 0.400 | 5 |
| Employee Knowledge & Skills | 1.000 | 0.023 | 0.045 | 43 |
| | 0.607 | 0.395 | 0.497 | 43 |
| | 0.700 | 0.488 | 0.575 | 43 |
| Handling | 0.742 | 0.377 | 0.500 | 61 |
| | 0.579 | 0.541 | 0.559 | 61 |
| | 0.544 | 0.607 | 0.574 | 61 |
| Information Provision | 0.472 | 0.970 | 0.635 | 202 |
| | 0.682 | 0.901 | 0.776 | 202 |
| | 0.795 | 0.861 | 0.827 | 202 |
| No topic found | 1.000 | 0.031 | 0.061 | 32 |
| | 0.636 | 0.219 | 0.326 | 32 |
| | 0.536 | 0.469 | 0.500 | 32 |
| Overall Experience | 0.857 | 0.400 | 0.545 | 60 |
| | 0.667 | 0.667 | 0.667 | 60 |
| | 0.639 | 0.767 | 0.697 | 60 |
| Price and Quality | 1.000 | 0.000 | 0.000 | 19 |
| | 0.625 | 0.263 | 0.370 | 19 |
| | 0.417 | 0.263 | 0.323 | 19 |
| Processes | 0.769 | 0.179 | 0.290 | 56 |
| | 0.692 | 0.482 | 0.568 | 56 |
| | 0.692 | 0.482 | 0.568 | 56 |
| Quality and Offering | 1.000 | 0.000 | 0.000 | 5 |
| | 1.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| accuracy | | | 0.544 | 607 |
| | | | 0.671 | 607 |
| | | | 0.697 | 607 |
| macro avg | 0.861 | 0.240 | 0.248 | 607 |
| | 0.640 | 0.437 | 0.469 | 607 |
| | 0.550 | 0.516 | 0.527 | 607 |
| weighted avg | 0.695 | 0.544 | 0.470 | 607 |
| | 0.665 | 0.671 | 0.646 | 607 |
| | 0.687 | 0.697 | 0.687 | 607 |

Table A.4: Overview results traditional machine learning on the 10% back-translated dataset, within a cell from top to bottom: Naive Bayes, Logistic Regression, and Support Vector machine

### A.1.5 Back-translation 20%

| Topic | Precision | Recall | F-Score | Support |
|---|---|---|---|---|
| Digital Options | 1.000 | 0.000 | 0.000 | 11 |
| | 0.857 | 0.545 | 0.667 | 11 |
| | 0.600 | 0.545 | 0.571 | 11 |
| Employee Attitude and Behavior | 0.627 | 0.655 | 0.641 | 113 |
| | 0.701 | 0.788 | 0.742 | 113 |
| | 0.733 | 0.779 | 0.755 | 113 |
| Employee Contact | 1.000 | 0.000 | 0.000 | 5 |
| | 1.000 | 0.000 | 0.000 | 5 |
| | 0.400 | 0.400 | 0.400 | 5 |
| Employee Knowledge & Skills | 1.000 | 0.023 | 0.045 | 43 |
| | 0.607 | 0.395 | 0.479 | 43 |
| | 0.606 | 0.465 | 0.526 | 43 |
| Handling | 0.765 | 0.426 | 0.547 | 61 |
| | 0.576 | 0.557 | 0.567 | 61 |
| | 0.544 | 0.607 | 0.574 | 61 |
| Information Provision | 0.476 | 0.970 | 0.638 | 202 |
| | 0.684 | 0.891 | 0.774 | 202 |
| | 0.785 | 0.861 | 0.817 | 202 |
| No topic found | 1.000 | 0.031 | 0.061 | 32 |
| | 0.500 | 0.156 | 0.238 | 32 |
| | 0.484 | 0.469 | 0.476 | 32 |
| Overall Experience | 0.857 | 0.400 | 0.545 | 60 |
| | 0.662 | 0.717 | 0.688 | 60 |
| | 0.672 | 0.750 | 0.709 | 60 |
| Price and Quality | 1.000 | 0.000 | 0.000 | 19 |
| | 0.600 | 0.316 | 0.414 | 19 |
| | 0.500 | 0.263 | 0.345 | 19 |
| Processes | 0.769 | 0.179 | 0.290 | 56 |
| | 0.711 | 0.482 | 0.574 | 56 |
| | 0.628 | 0.482 | 0.545 | 56 |
| Quality and Offering | 1.000 | 0.000 | 0.000 | 5 |
| | 1.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| accuracy | | | 0.547 | 607 |
| | | | 0.671 | 607 |
| | | | 0.687 | 607 |
| macro avg | 0.863 | 0.244 | 0.252 | 607 |
| | 0.718 | 0.441 | 0.467 | 607 |
| | 0.541 | 0.510 | 0.520 | 607 |
| weighted avg | 0.697 | 0.547 | 0.474 | 607 |
| | 0.667 | 0.671 | 0.645 | 607 |
| | 0.675 | 0.687 | 0.677 | 607 |

Table A.5: Overview results traditional machine learning on the 20% back-translated dataset, within a cell from top to bottom: Naive Bayes, Logistic Regression, and Support Vector machine

### A.1.6 Undersampled Dataset

| Topic | Precision | Recall | F-Score | Support |
|---|---|---|---|---|
| Digital Options | 0.104 | 0.455 | 0.169 | 11 |
| | 0.214 | 0.545 | 0.308 | 11 |
| | 0.182 | 0.545 | 0.273 | 11 |
| Employee Attitude and Behavior | 0.649 | 0.327 | 0.435 | 113 |
| | 0.658 | 0.442 | 0.529 | 113 |
| | 0.676 | 0.425 | 0.522 | 113 |
| Employee Contact | 0.037 | 0.200 | 0.062 | 5 |
| | 0.056 | 0.200 | 0.087 | 5 |
| | 0.074 | 0.400 | 0.125 | 5 |
| Employee Knowledge & Skills | 0.281 | 0.419 | 0.336 | 43 |
| | 0.236 | 0.302 | 0.265 | 43 |
| | 0.309 | 0.395 | 0.347 | 43 |
| Handling | 0.315 | 0.557 | 0.402 | 61 |
| | 0.426 | 0.475 | 0.450 | 61 |
| | 0.443 | 0.508 | 0.473 | 61 |
| Information Provision | 0.726 | 0.421 | 0.533 | 202 |
| | 0.737 | 0.431 | 0.544 | 202 |
| | 0.721 | 0.436 | 0.543 | 202 |
| No topic found | 0.200 | 0.250 | 0.222 | 32 |
| | 0.159 | 0.344 | 0.218 | 32 |
| | 0.149 | 0.312 | 0.202 | 32 |
| Overall Experience | 0.404 | 0.350 | 0.375 | 60 |
| | 0.400 | 0.433 | 0.416 | 60 |
| | 0.379 | 0.417 | 0.397 | 60 |
| Price and Quality | 0.348 | 0.421 | 0.381 | 19 |
| | 0.310 | 0.474 | 0.375 | 19 |
| | 0.320 | 0.421 | 0.364 | 19 |
| Processes | 0.388 | 0.339 | 0.362 | 56 |
| | 0.333 | 0.304 | 0.318 | 56 |
| | 0.375 | 0.321 | 0.346 | 56 |
| Quality and Offering | 0.045 | 0.200 | 0.074 | 5 |
| | 0.033 | 0.200 | 0.057 | 5 |
| | 0.043 | 0.200 | 0.071 | 5 |
| accuracy | | | 0.390 | 607 |
| | | | 0.412 | 607 |
| | | | 0.418 | 607 |
| macro avg | 0.318 | 0.358 | 0.305 | 607 |
| | 0.324 | 0.377 | 0.342 | 607 |
| | 0.334 | 0.398 | 0.333 | 607 |
| weighted avg | 0.514 | 0.390 | 0.421 | 607 |
| | 0.520 | 0.412 | 0.444 | 607 |
| | 0.526 | 0.418 | 0.450 | 607 |

Table A.6: Overview results traditional machine learning on the undersampled dataset, within a cell from top to bottom: Naive Bayes, Logistic Regression, and Support Vector machine

## A.2 Embeddings

### A.2.1 Original Dataset

| | Finance | | | Google | | | |
|---|---|---|---|---|---|---|---|
| Topic | Precision | Recall | F-Score | Precision | Recall | F-Score | S |
| Digital Options | 0.429 | 0.273 | 0.333 | 0.500 | 0.091 | 0.154 | 11 |
| | 0.333 | 0.273 | 0.300 | 0.333 | 0.091 | 0.143 | 11 |
| Employee Attitude and Behavior | 0.604 | 0.487 | 0.539 | 0.625 | 0.442 | 0.518 | 113 |
| | 0.567 | 0.487 | 0.524 | 0.649 | 0.442 | 0.526 | 113 |
| Employee Contact | 1.000 | 0.000 | 0.000 | 1.000 | 0.200 | 0.333 | 5 |
| | 0.000 | 0.000 | 0.000 | 1.000 | 0.200 | 0.333 | 5 |
| Employee Knowledge & Skills | 0.350 | 0.163 | 0.222 | 0.600 | 0.209 | 0.310 | 43 |
| | 0.375 | 0.140 | 0.203 | 0.619 | 0.302 | 0.406 | 43 |
| Handling | 0.568 | 0.410 | 0.470 | 0.628 | 0.443 | 0.519 | 61 |
| | 0.578 | 0.426 | 0.491 | 0.587 | 0.443 | 0.505 | 61 |
| Information Provision | 0.503 | 0.876 | 0.639 | 0.505 | 0.911 | 0.650 | 202 |
| | 0.504 | 0.871 | 0.639 | 0.527 | 0.911 | 0.668 | 202 |
| No topic found | 0.222 | 0.062 | 0.098 | 0.538 | 0.219 | 0.311 | 32 |
| | 0.143 | 0.031 | 0.051 | 0.429 | 0.281 | 0.340 | 32 |
| Overall Experience | 0.556 | 0.500 | 0.526 | 0.610 | 0.600 | 0.605 | 60 |
| | 0.564 | 0.517 | 0.539 | 0.593 | 0.583 | 0.588 | 60 |
| Price and Quality | 0.500 | 0.105 | 0.174 | 0.667 | 0.105 | 0.182 | 19 |
| | 0. 500 | 0.105 | 0.174 | 0.800 | 0.211 | 0.333 | 19 |
| Processes | 0.480 | 0.214 | 0.296 | 0.704 | 0.339 | 0.458 | 56 |
| | 0.500 | 0.214 | 0.300 | 0.680 | 0.304 | 0.420 | 56 |
| Quality and Offering | 1.000 | 0.200 | 0.333 | 1.000 | 0.000 | 0.000 | 5 |
| | 1.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 5 |
| accuracy | | | 0.517 | | | 0.554 | 607 |
| | | | 0.514 | | | 0.562 | 607 |
| macro avg | 0.565 | 0.299 | 0.331 | 0.671 | 0.324 | 0.367 | 607 |
| | 0.460 | 0.279 | 0.293 | 0.656 | 0.343 | 0.530 | 607 |
| weighted avg | 0.513 | 0.517 | 0.475 | 0.590 | 0.554 | 0.517 | 607 |
| | 0.497 | 0.514 | 0.468 | 0.591 | 0.562 | 0.530 | 607 |

Table A.7: Overview results traditional machine learning with embeddings on the original dataset, within a cell from top to bottom: Logistic Regression, and Support Vector machine

### A.2.2 Merged Dataset

| | Finance | | | Google | | | |
|---|---|---|---|---|---|---|---|
| Topic | Precision | Recall | F-Score | Precision | Recall | F-Score | S |
| Digital Options | 0.500 | 0.273 | 0.353 | 0.500 | 0.091 | 0.154 | 11 |
| | 0.333 | 0.273 | 0.300 | 1.000 | 0.091 | 0.167 | 11 |
| Employee | 0.453 | 0.652 | 0.543 | 0.599 | 0.528 | 0.561 | 161 |
| | 0.450 | 0.677 | 0.541 | 0.622 | 0.522 | 0.568 | 161 |
| Handling | 0.545 | 0.393 | 0.457 | 0.639 | 0.377 | 0.474 | 61 |
| | 0.571 | 0.393 | 0.466 | 0.581 | 0.410 | 0.481 | 61 |
| Information Provision | 0.563 | 0.663 | 0.609 | 0.529 | 0.856 | 0.654 | 202 |
| | 0.557 | 0.634 | 0.593 | 0.542 | 0.861 | 0.665 | 202 |
| No topic found | 0.333 | 0.062 | 0.105 | 0.545 | 0.188 | 0.279 | 32 |
| | 0.000 | 0.000 | 0.000 | 0.444 | 0.250 | 0.320 | 32 |
| Overall Experience | 0.529 | 0.450 | 0.486 | 0.607 | 0.567 | 0.586 | 60 |
| | 0.566 | 0.500 | 0.531 | 0.614 | 0.583 | 0.598 | 60 |
| Price and Quality | 0.600 | 0.125 | 0.207 | 0.833 | 0.208 | 0.333 | 24 |
| | 0.600 | 0.125 | 0.207 | 0.857 | 0.250 | 0.387 | 24 |
| Processes | 0.440 | 0.196 | 0.272 | 0.704 | 0.339 | 0.458 | 56 |
| | 0.455 | 0.179 | 0.256 | 0.680 | 0.304 | 0.420 | 56 |
| accuracy | | | 0.509 | | | 0.570 | 607 |
| | | | 0.506 | | | 0.577 | 607 |
| macro avg | 0.495 | 0.352 | 0.378 | 0.620 | 0.394 | 0.437 | 607 |
| | 0.442 | 0.348 | 0.362 | 0.668 | 0.409 | 0.451 | 607 |
| weighted avg | 0.506 | 0.509 | 0.484 | 0.595 | 0.570 | 0.545 | 607 |
| | 0.490 | 0.506 | 0.477 | 0.603 | 0.577 | 0.553 | 607 |

Table A.8: Overview results traditional machine learning with embeddings after merging topics with an overlapping content, within a cell from top to bottom: Logistic Regression, and Support Vector machine

### A.2.3 Merged "Other" Dataset

| | Finance | | | Google | | | |
|---|---|---|---|---|---|---|---|
| Topic | Precision | Recall | F-Score | Precision | Recall | F-Score | S |
| Employee Attitude and Behavior | 0.604 | 0.487 | 0.539 | 0.636 | 0.434 | 0.516 | 113 |
| | 0.567 | 0.487 | 0.524 | 0.658 | 0.442 | 0.529 | 113 |
| Employee Knowledge & Skills | 0.333 | 0.163 | 0.219 | 0.529 | 0.209 | 0.300 | 43 |
| | 0.375 | 0.140 | 0.203 | 0.619 | 0.302 | 0.406 | 43 |
| Handling | 0.556 | 0.410 | 0.472 | 0.628 | 0.443 | 0.519 | 61 |
| | 0.542 | 0.426 | 0.477 | 0.587 | 0.443 | 0.505 | 61 |
| Information Provision | 0.504 | 0.876 | 0.640 | 0.506 | 0.906 | 0.649 | 202 |
| | 0.503 | 0.866 | 0.636 | 0.527 | 0.906 | 0.667 | 202 |
| No topic found | 0.250 | 0.062 | 0.100 | 0.583 | 0.219 | 0.318 | 32 |
| | 0.167 | 0.031 | 0.053 | 0.450 | 0.281 | 0.346 | 32 |
| Other | 0.500 | 0.175 | 0.259 | 0.750 | 0.225 | 0.346 | 40 |
| | 0.500 | 0.150 | 0.231 | 0.667 | 0.250 | 0.364 | 40 |
| Overall Experience | 0.558 | 0.483 | 0.518 | 0.621 | 0.600 | 0.610 | 60 |
| | 0.564 | 0.517 | 0.390 | 0.603 | 0.583 | 0.593 | 60 |
| Processes | 0.480 | 0.214 | 0.296 | 0.731 | 0.339 | 0.463 | 56 |
| | 0.520 | 0.232 | 0.321 | 0.708 | 0.304 | 0.425 | 56 |
| accuracy | | | 0.517 | | | 0.558 | 607 |
| | | | 0.516 | | | 0.567 | 607 |
| macro avg | 0.473 | 0.359 | 0.380 | 0.623 | 0.422 | 0.465 | 607 |
| | 0.467 | 0.356 | 0.373 | 0.602 | 0.439 | 0.479 | 607 |
| weighted avg | 0.505 | 0.517 | 0.477 | 0.596 | 0.558 | 0.528 | 607 |
| | 0.499 | 0.516 | 0.473 | 0.593 | 0.567 | 0.540 | 607 |

Table A.9: Overview results traditional machine learning with embeddings after merging underrepresented topics to a topic called "Other", within a cell from top to bottom: Logistic Regression, and Support Vector machine

### A.2.4 Back-translation 10%

| | Finance | | | Google | | | |
|---|---|---|---|---|---|---|---|
| Topic | Precision | Recall | F-Score | Precision | Recall | F-Score | S |
| Digital Options | 0.500 | 0.273 | 0.353 | 0.333 | 0.091 | 0.143 | 11 |
| | 0.333 | 0.273 | 0.300 | 0.250 | 0.091 | 0.133 | 11 |
| Employee Attitude and Behavior | 0.604 | 0.487 | 0.539 | 0.649 | 0.442 | 0.526 | 113 |
| | 0.602 | 0.496 | 0.544 | 0.632 | 0.425 | 0.508 | 113 |
| Employee Contact | 1.000 | 0.000 | 0.000 | 1.000 | 0.200 | 0.333 | 5 |
| | 0.000 | 0.000 | 0.000 | 0.500 | 0.200 | 0.286 | 5 |
| Employee Knowledge & Skills | 0.180 | 0.163 | 0.215 | 0.600 | 0.209 | 0.310 | 43 |
| | 0.500 | 0.163 | 0.246 | 0.524 | 0.256 | 0.344 | 43 |
| Handling | 0.558 | 0.393 | 0.462 | 0.651 | 0.459 | 0.538 | 61 |
| | 0.551 | 0.443 | 0.491 | 0.643 | 0.443 | 0.524 | 61 |
| Information Provision | 0.500 | 0.876 | 0.637 | 0.501 | 0.911 | 0.647 | 202 |
| | 0.507 | 0.886 | 0.645 | 0.532 | 0.911 | 0.672 | 202 |
| No topic found | 0.286 | 0.062 | 0.103 | 0.500 | 0.219 | 0.304 | 32 |
| | 0.167 | 0.031 | 0.053 | 0.450 | 0.281 | 0.346 | 32 |
| Overall Experience | 0.529 | 0.450 | 0.486 | 0.610 | 0.600 | 0.605 | 60 |
| | 0.544 | 0.517 | 0.530 | 0.613 | 0.633 | 0.623 | 60 |
| Price and Quality | 0.400 | 0.105 | 0.167 | 0.667 | 0.105 | 0.182 | 19 |
| | 0.400 | 0.105 | 0.167 | 0.800 | 0.211 | 0.333 | 19 |
| Processes | 0.429 | 0.214 | 0.286 | 0.680 | 0.304 | 0.420 | 56 |
| | 0.550 | 0.196 | 0.289 | 0.621 | 0.321 | 0.424 | 56 |
| Quality and Offering | 1.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 5 |
| | 1.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 5 |
| accuracy | | | 0.509 | | | 0.552 | 607 |
| | | | 0.522 | | | 0.562 | 607 |
| macro avg | 0.557 | 0.275 | 0.295 | 0.654 | 0.322 | 0.364 | 607 |
| | 0.469 | 0.283 | 0.297 | 0.597 | 0.343 | 0.381 | 607 |
| weighted avg | 0.503 | 0.509 | 0.465 | 0.588 | 0.552 | 0.515 | 607 |
| | 0.512 | 0.522 | 0.475 | 0.580 | 0.562 | 0.529 | 607 |

Table A.10: Overview results traditional machine learning with embeddings after back-translating 10% of each topic of the original dataset, within a cell from top to bottom: Logistic Regression, and Support Vector machine

### A.2.5 Back-translation 20%

| | Finance | | | Google | | | |
|---|---|---|---|---|---|---|---|
| Topic | Precision | Recall | F-Score | Precision | Recall | F-Score | S |
| Digital Options | 0.429 | 0.273 | 0.333 | 0.333 | 0.091 | 0.143 | 11 |
| | 0.375 | 0.273 | 0.316 | 0.200 | 0.091 | 0.125 | 11 |
| Employee Attitude and Behavior | 0.579 | 0.487 | 0.529 | 0.649 | 0.442 | 0.526 | 113 |
| | 0.577 | 0.496 | 0.533 | 0.681 | 0.434 | 0.530 | 113 |
| Employee Contact | 1.000 | 0.000 | 0.000 | 1.000 | 0.200 | 0.333 | 5 |
| | 1.000 | 0.000 | 0.000 | 0.500 | 0.200 | 0.286 | 5 |
| Employee Knowledge & Skills | 0.350 | 0.163 | 0.222 | 0.588 | 0.233 | 0.333 | 43 |
| | 0.412 | 0.163 | 0.233 | 0.542 | 0.302 | 0.388 | 43 |
| Handling | 0.556 | 0.410 | 0.472 | 0.643 | 0.443 | 0.524 | 61 |
| | 0.551 | 0.443 | 0.491 | 0.600 | 0.443 | 0.509 | 61 |
| Information Provision | 0.500 | 0.871 | 0.635 | 0.508 | 0.916 | 0.654 | 202 |
| | 0.513 | 0.881 | 0.648 | 0.530 | 0.906 | 0.669 | 202 |
| No topic found | 0.143 | 0.031 | 0.051 | 0.538 | 0.219 | 0.311 | 32 |
| | 0.200 | 0.031 | 0.054 | 0.364 | 0.250 | 0.296 | 32 |
| Overall Experience | 0.500 | 0.417 | 0.455 | 0.633 | 0.633 | 0.633 | 60 |
| | 0.554 | 0.517 | 0.534 | 0.638 | 0.617 | 0.627 | 60 |
| Price and Quality | 0.400 | 0.105 | 0.167 | 0.750 | 0.158 | 0.261 | 19 |
| | 0.400 | 0.105 | 0.167 | 0.833 | 0.263 | 0.400 | 19 |
| Processes | 0.462 | 0.214 | 0.293 | 0.692 | 0.321 | 0.439 | 56 |
| | 0.522 | 0.214 | 0.304 | 0.679 | 0.339 | 0.452 | 56 |
| Quality and Offering | 1.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 5 |
| | 1.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 5 |
| accuracy | | | 0.504 | | | 0.560 | 607 |
| | | | 0.522 | | | 0.565 | 607 |
| macro avg | 0.538 | 0.270 | 0.287 | 0.667 | 0.322 | 0.378 | 607 |
| | 0.555 | 0.284 | 0.298 | 0.597 | 0.350 | 0.389 | 607 |
| weighted avg | 0.491 | 0.504 | 0.459 | 0.597 | 0.560 | 0.525 | 607 |
| | 0.512 | 0.522 | 0.476 | 0.589 | 0.565 | 0.536 | 607 |

Table A.11: Overview results traditional machine learning with embeddings after back-translating 20% of each topic of the original dataset, within a cell from top to bottom: Logistic Regression, and Support Vector machine

### A.2.6 Undersampling

| | Finance | | | Google | | | |
|---|---|---|---|---|---|---|---|
| Topic | Precision | Recall | F-Score | Precision | Recall | F-Score | S |
| Digital Options | 0.043 | 0.091 | 0.059 | 0.051 | 0.909 | 0.096 | 11 |
| | 0.057 | 0.182 | 0.087 | 0.037 | 0.455 | 0.068 | 11 |
| Employee Attitude and Behavior | 0.500 | 0.212 | 0.298 | 0.714 | 0.265 | 0.387 | 113 |
| | 0.436 | 0.212 | 0.286 | 0.667 | 0.212 | 0.322 | 113 |
| Employee Contact | 0.056 | 0.200 | 0.087 | 0.062 | 0.200 | 0.095 | 5 |
| | 0.053 | 0.200 | 0.085 | 0.071 | 0.200 | 0.105 | 5 |
| Employee Knowledge & Skills | 0.184 | 0.209 | 0.196 | 0.310 | 0.302 | 0.306 | 43 |
| | 0.182 | 0.233 | 0.204 | 0.310 | 0.302 | 0.306 | 43 |
| Handling | 0.393 | 0.361 | 0.376 | 0.568 | 0.344 | 0.429 | 61 |
| | 0.464 | 0.426 | 0.444 | 0.438 | 0.344 | 0.385 | 61 |
| Information Provision | 0.639 | 0.228 | 0.336 | 0.646 | 0.307 | 0.416 | 202 |
| | 0.639 | 0.228 | 0.336 | 0.627 | 0.342 | 0.442 | 202 |
| No topic found | 0.103 | 0.188 | 0.133 | 0.208 | 0.344 | 0.259 | 32 |
| | 0.093 | 0.156 | 0.116 | 0.218 | 0.375 | 0.276 | 32 |
| Overall Experience | 0.302 | 0.217 | 0.252 | 0.370 | 0.283 | 0.321 | 60 |
| | 0.327 | 0.267 | 0.294 | 0.367 | 0.300 | 0.330 | 60 |
| Price and Quality | 0.308 | 0.421 | 0.356 | 0.333 | 0.368 | 0.350 | 19 |
| | 0.164 | 0.474 | 0.243 | 0.179 | 0.368 | 0.241 | 19 |
| Processes | 0.110 | 0.393 | 0.172 | 0.463 | 0.339 | 0.392 | 56 |
| | 0.135 | 0.339 | 0.193 | 0.404 | 0.375 | 0.389 | 56 |
| Quality and Offering | 0.071 | 0.200 | 0.105 | 0.000 | 0.000 | 0.000 | 5 |
| | 0.062 | 0.200 | 0.095 | 0.037 | 0.200 | 0.062 | 5 |
| accuracy | | | 0.252 | | | 0.315 | 607 |
| | | | 0.262 | | | 0.316 | 607 |
| macro avg | 0.246 | 0.247 | 0.215 | 0.339 | 0.333 | 0.277 | 607 |
| | 0.237 | 0.265 | 0.217 | 0.305 | 0.316 | 0.266 | 607 |
| weighted avg | 0.415 | 0.252 | 0.281 | 0.529 | 0.315 | 0.370 | 607 |
| | 0.410 | 0.262 | 0.288 | 0.491 | 0.316 | 0.361 | 607 |

Table A.12: Overview results traditional machine learning with embeddings after undersampling each topic of the original dataset, within a cell from top to bottom: Logistic Regression, and Support Vector machine

# Appendix B

# Results Transfer Learning

## B.1 Original Dataset

| Topic | Precision | Recall | F-Score | Support |
|---|---|---|---|---|
| Digital Options | 0.500 | 0.545 | 0.522 | 11 |
| | 0.474 | 0.818 | 0.600 | 11 |
| | 0.333 | 0.455 | 0.385 | 11 |
| Employee Attitude and Behavior | 0.838 | 0.735 | 0.783 | 113 |
| | 0.817 | 0.788 | 0.802 | 113 |
| | 0.81 | 0.832 | 0.821 | 113 |
| Employee Contact | 0.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| Employee Knowledge & Skills | 0.545 | 0.419 | 0.474 | 43 |
| | 0.625 | 0.465 | 0.533 | 43 |
| | 0.600 | 0.488 | 0.538 | 43 |
| Handling | 0.741 | 0.705 | 0.723 | 61 |
| | 0.732 | 0.672 | 0.701 | 61 |
| | 0.667 | 0.721 | 0.693 | 61 |
| Information Provision | 0.774 | 0.866 | 0.818 | 202 |
| | 0.797 | 0.876 | 0.835 | 202 |
| | 0.793 | 0.851 | 0.821 | 202 |
| No topic found | 0.429 | 0.469 | 0.448 | 32 |
| | 0.462 | 0.375 | 0.414 | 32 |
| | 0.421 | 0.500 | 0.457 | 32 |
| Overall Experience | 0.603 | 0.783 | 0.681 | 60 |
| | 0.622 | 0.767 | 0.687 | 60 |
| | 0.698 | 0.733 | 0.715 | 60 |
| Price and Quality | 0.600 | 0.474 | 0.529 | 19 |
| | 0.529 | 0.474 | 0.500 | 19 |
| | 0.769 | 0.526 | 0.625 | 19 |
| Processes | 0.745 | 0.679 | 0.710 | 56 |
| | 0.769 | 0.714 | 0.741 | 56 |
| | 0.818 | 0.643 | 0.72 | 56 |
| Quality and Offering | 0.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| accuracy | | | 0.715 | 607 |
| | | | 0.73 | 607 |
| | | | 0.728 | 607 |
| macro avg | 0.525 | 0.516 | 0.517 | 607 |
| | 0.530 | 0.541 | 0.528 | 607 |
| | 0.537 | 0.523 | 0.525 | 607 |
| weighted avg | 0.706 | 0.715 | 0.707 | 607 |
| | 0.717 | 0.730 | 0.720 | 607 |
| | 0.721 | 0.728 | 0.722 | 607 |

Table B.1: Overview results transfer learning on the original dataset, within a cell from top to bottom: BERT, RoBERTa, DistilBERT

## B.2 Merged Dataset

| Topic | Precision | Recall | F-Score | Support |
|---|---|---|---|---|
| Digital Options | 0.533 | 0.727 | 0.615 | 11 |
| | 0.571 | 0.727 | 0.640 | 11 |
| | 0.455 | 0.455 | 0.455 | 11 |
| Employee | 0.862 | 0.776 | 0.817 | 161 |
| | 0.848 | 0.764 | 0.804 | 161 |
| | 0.836 | 0.789 | 0.812 | 161 |
| Handling | 0.643 | 0.738 | 0.687 | 61 |
| | 0.712 | 0.770 | 0.740 | 61 |
| | 0.651 | 0.672 | 0.661 | 61 |
| Information Provision | 0.808 | 0.851 | 0.829 | 202 |
| | 0.788 | 0.881 | 0.832 | 202 |
| | 0.805 | 0.881 | 0.842 | 202 |
| No topic found | 0.483 | 0.438 | 0.459 | 32 |
| | 0.500 | 0.375 | 0.429 | 32 |
| | 0.474 | 0.281 | 0.353 | 32 |
| Overall Experience | 0.677 | 0.700 | 0.689 | 60 |
| | 0.656 | 0.700 | 0.677 | 60 |
| | 0.648 | 0.767 | 0.702 | 60 |
| Price and Quality | 0.632 | 0.500 | 0.558 | 19 |
| | 0.522 | 0.500 | 0.511 | 19 |
| | 0.632 | 0.500 | 0.558 | 19 |
| Processes | 0.648 | 0.625 | 0.636 | 56 |
| | 0.800 | 0.643 | 0.713 | 56 |
| | 0.745 | 0.679 | 0.710 | 56 |
| accuracy | | | 0.746 | 607 |
| | | | 0.755 | 607 |
| | | | 0.751 | 607 |
| macro avg | 0.661 | 0.669 | 0.661 | 607 |
| | 0.675 | 0.670 | 0.668 | 607 |
| | 0.656 | 0.628 | 0.637 | 607 |
| weighted avg | 0.749 | 0.746 | 0.746 | 607 |
| | 0.755 | 0.755 | 0.752 | 607 |
| | 0.746 | 0.751 | 0.746 | 607 |

Table B.2: Overview results transfer learning on the merged dataset, within a cell from top to bottom: BERT, RoBERTa, DistilBERT

## B.3 Merged "Other" Dataset

| Topic | Precision | Recall | F-Score | Support |
|---|---|---|---|---|
| Employee Attitude and Behavior | 0.835 | 0.761 | 0.796 | 113 |
| | 0.820 | 0.805 | 0.812 | 113 |
| | 0.817 | 0.788 | 0.802 | 113 |
| Employee Knowledge & Skills | 0.625 | 0.465 | 0.533 | 43 |
| | 0.676 | 0.581 | 0.625 | 43 |
| | 0.562 | 0.419 | 0.480 | 43 |
| Handling | 0.657 | 0.721 | 0.688 | 61 |
| | 0.662 | 0.738 | 0.698 | 61 |
| | 0.688 | 0.721 | 0.704 | 61 |
| Information Provision | 0.786 | 0.856 | 0.820 | 202 |
| | 0.808 | 0.856 | 0.832 | 202 |
| | 0.781 | 0.866 | 0.822 | 202 |
| No topic found | 0.559 | 0.594 | 0.576 | 32 |
| | 0.474 | 0.281 | 0.353 | 32 |
| | 0.500 | 0.406 | 0.448 | 32 |
| Overall Experience | 0.688 | 0.733 | 0.710 | 60 |
| | 0.642 | 0.717 | 0.677 | 60 |
| | 0.657 | 0.733 | 0.693 | 60 |
| Processes | 0.739 | 0.607 | 0.667 | 56 |
| | 0.771 | 0.661 | 0.712 | 56 |
| | 0.702 | 0.589 | 0.641 | 56 |
| Other | 0.537 | 0.550 | 0.543 | 40 |
| | 0.628 | 0.675 | 0.651 | 40 |
| | 0.632 | 0.600 | 0.615 | 40 |
| accuracy | | | 0.728 | 607 |
| | | | 0.741 | 607 |
| | | | 0.725 | 607 |
| macro avg | 0.678 | 0.661 | 0.667 | 607 |
| | 0.685 | 0.664 | 0.670 | 607 |
| | 0.667 | 0.640 | 0.651 | 607 |
| weighted avg | 0.728 | 0.728 | 0.726 | 607 |
| | 0.737 | 0.741 | 0.736 | 607 |
| | 0.719 | 0.725 | 0.719 | 607 |

Table B.3: Overview results transfer learning on the other dataset, within a cell from top to bottom: BERT, RoBERTa, DistilBERT

## B.4 Back-translation 10%

| Topic | Precision | Recall | F-Score | Support |
|---|---|---|---|---|
| Digital Options | 0.429 | 0.545 | 0.480 | 11 |
| | 0.500 | 0.727 | 0.593 | 11 |
| | 0.455 | 0.455 | 0.455 | 11 |
| Employee Attitude and Behavior | 0.832 | 0.788 | 0.809 | 113 |
| | 0.841 | 0.796 | 0.818 | 113 |
| | 0.791 | 0.770 | 0.780 | 113 |
| Employee Contact | 0.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| Employee Knowledge & Skills | 0.600 | 0.419 | 0.493 | 43 |
| | 0.639 | 0.535 | 0.582 | 43 |
| | 0.538 | 0.488 | 0.512 | 43 |
| Handling | 0.768 | 0.705 | 0.735 | 61 |
| | 0.708 | 0.754 | 0.730 | 61 |
| | 0.689 | 0.689 | 0.689 | 61 |
| Information Provision | 0.784 | 0.881 | 0.830 | 202 |
| | 0.805 | 0.881 | 0.842 | 202 |
| | 0.771 | 0.866 | 0.816 | 202 |
| No topic found | 0.484 | 0.469 | 0.476 | 32 |
| | 0.500 | 0.281 | 0.360 | 32 |
| | 0.448 | 0.406 | 0.426 | 32 |
| Overall Experience | 0.625 | 0.750 | 0.682 | 60 |
| | 0.641 | 0.683 | 0.661 | 60 |
| | 0.691 | 0.783 | 0.734 | 60 |
| Price and Quality | 0.600 | 0.474 | 0.529 | 19 |
| | 0.360 | 0.474 | 0.409 | 19 |
| | 0.667 | 0.526 | 0.588 | 19 |
| Processes | 0.727 | 0.714 | 0.721 | 56 |
| | 0.655 | 0.643 | 0.649 | 56 |
| | 0.723 | 0.607 | 0.660 | 56 |
| Quality and Offering | 0.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| accuracy | | | 0.730 | 607 |
| | | | 0.725 | 607 |
| | | | 0.715 | 607 |
| macro avg | 0.532 | 0.522 | 0.523 | 607 |
| | 0.513 | 0.525 | 0.513 | 607 |
| | 0.525 | 0.508 | 0.515 | 607 |
| weighted avg | 0.716 | 0.730 | 0.720 | 607 |
| | 0.711 | 0.725 | 0.715 | 607 |
| | 0.699 | 0.715 | 0.705 | 607 |

Table B.4: Overview results transfer learning on the 10% back-translated dataset, within a cell from top to bottom: BERT, RoBERTa, and DistilBERT

## B.5 Back-translation 20%

| Topic | Precision | Recall | F-Score | Support |
|---|---|---|---|---|
| Digital Options | 0.438 | 0.636 | 0.519 | 11 |
| | 0.450 | 0.818 | 0.581 | 11 |
| | 0.455 | 0.455 | 0.455 | 11 |
| Employee Attitude and Behavior | 0.853 | 0.770 | 0.809 | 113 |
| | 0.806 | 0.770 | 0.787 | 113 |
| | 0.811 | 0.796 | 0.804 | 113 |
| Employee Contact | 0.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| Employee Knowledge & Skills | 0.571 | 0.465 | 0.513 | 43 |
| | 0.625 | 0.465 | 0.533 | 43 |
| | 0.636 | 0.488 | 0.553 | 43 |
| Handling | 0.726 | 0.738 | 0.732 | 61 |
| | 0.723 | 0.770 | 0.746 | 61 |
| | 0.653 | 0.770 | 0.707 | 61 |
| Information Provision | 0.795 | 0.866 | 0.829 | 202 |
| | 0.802 | 0.881 | 0.840 | 202 |
| | 0.804 | 0.851 | 0.827 | 202 |
| No topic found | 0.471 | 0.500 | 0.485 | 32 |
| | 0.522 | 0.375 | 0.436 | 32 |
| | 0.400 | 0.438 | 0.418 | 32 |
| Overall Experience | 0.647 | 0.733 | 0.688 | 60 |
| | 0.614 | 0.717 | 0.662 | 60 |
| | 0.638 | 0.733 | 0.682 | 60 |
| Price and Quality | 0.571 | 0.421 | 0.485 | 19 |
| | 0.474 | 0.474 | 0.474 | 19 |
| | 0.692 | 0.474 | 0.562 | 19 |
| Processes | 0.714 | 0.714 | 0.714 | 56 |
| | 0.729 | 0.625 | 0.673 | 56 |
| | 0.714 | 0.625 | 0.667 | 56 |
| Quality and Offering | 0.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| accuracy | | | 0.728 | 607 |
| | | | 0.725 | 607 |
| | | | 0.720 | 607 |
| macro avg | 0.526 | 0.531 | 0.525 | 607 |
| | 0.522 | 0.536 | 0.521 | 607 |
| | 0.528 | 0.512 | 0.516 | 607 |
| weighted avg | 0.717 | 0.728 | 0.721 | 607 |
| | 0.712 | 0.725 | 0.715 | 607 |
| | 0.709 | 0.720 | 0.712 | 607 |

Table B.5: Overview results transfer learning on the 20% back-translated dataset, within a cell from top to bottom: BERT, RoBERTa, and DistilBERT

## B.6 Undersampled Dataset

| Topic | Precision | Recall | F-Score | Support |
|---|---|---|---|---|
| Digital Options | 0.208 | 0.455 | 0.286 | 11 |
| | 0.000 | 0.000 | 0.000 | 11 |
| | 0.250 | 0.091 | 0.133 | 11 |
| Employee Attitude and Behavior | 0.273 | 0.080 | 0.123 | 113 |
| | 0.000 | 0.000 | 0.000 | 113 |
| | 1.000 | 0.009 | 0.018 | 113 |
| Employee Contact | 0.000 | 0.000 | 0.000 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| | 0.012 | 0.800 | 0.024 | 5 |
| Employee Knowledge & Skills | 0.000 | 0.000 | 0.000 | 43 |
| | 0.000 | 0.000 | 0.000 | 43 |
| | 0.299 | 0.186 | 0.205 | 43 |
| Handling | 0.000 | 0.000 | 0.000 | 61 |
| | 0.000 | 0.000 | 0.000 | 61 |
| | 0.375 | 0.098 | 0.156 | 61 |
| Information Provision | 0.200 | 0.005 | 0.010 | 202 |
| | 0.352 | 0.980 | 0.518 | 202 |
| | 0.000 | 0.000 | 0.000 | 202 |
| No topic found | 0.080 | 0.438 | 0.136 | 32 |
| | 0.103 | 0.125 | 0.113 | 32 |
| | 0.000 | 0.000 | 0.000 | 32 |
| Overall Experience | 0.000 | 0.000 | 0.000 | 60 |
| | 0.000 | 0.000 | 0.000 | 60 |
| | 0.000 | 0.000 | 0.000 | 60 |
| Price and Quality | 0.036 | 0.438 | 0.136 | 19 |
| | 0.000 | 0.000 | 0.000 | 19 |
| | 0.100 | 0.316 | 0.152 | 19 |
| Processes | 0.171 | 0.214 | 0.190 | 56 |
| | 0.250 | 0.018 | 0.033 | 56 |
| | 0.152 | 0.339 | 0.210 | 56 |
| Quality and Offering | 0.009 | 0.400 | 0.017 | 5 |
| | 0.000 | 0.000 | 0.000 | 5 |
| | 0.050 | 0.400 | 0.089 | 5 |
| accuracy | | | 0.072 | 607 |
| | | | 0.334 | 607 |
| | | | 0.288 | 607 |
| macro avg | 0.089 | 0.149 | 0.073 | 607 |
| | 0.064 | 0.102 | 0.060 | 607 |
| | 0.288 | 0.205 | 0.093 | 607 |
| weighted avg | 0.142 | 0.072 | 0.058 | 607 |
| | 0.146 | 0.334 | 0.182 | 607 |
| | 0.361 | 0.079 | 0.064 | 607 |

Table B.6: Overview results transfer learning on the undersampled dataset, within a cell from top to bottom: BERT, RoBERTa, DistilBERT

# Bibliography


C. C. Aggarwal and C. Zhai. A survey of text classification algorithms. In *Mining text data*, pages 163–222. Springer, 2012.

N. A.-R. Al-Serw. Undersampling and oversampling: An old and a new approach, Feb 2021. URL `https://medium.com/analytics-vidhya/undersampling-and-oversampling-an-old-and-a-new-approach-4f984a0e8392`.

J. Alammar. The illustrated bert, elmo, and co.(how nlp cracked transfer learning)(2018), 2018a.

J. Alammar. The illustrated transformer, Jun 2018b. URL `https://jalammar.github.io/illustrated-transformer/`.

N. Aslam, W. Y. Ramay, K. Xia, and N. Sarwar. Convolutional neural network based classification of app reviews. *IEEE Access*, 8:185619–185628, 2020.

J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization, 2016. URL `https://arxiv.org/abs/1607.06450`.

T. Bayes. Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s. *Philosophical transactions of the Royal Society of London*, (53):370–418, 1763.

E. M. Bender and B. Friedman. Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science. *Transactions of the Association for Computational Linguistics*, 6:587–604, 12 2018. ISSN 2307-387X. doi: 10.1162/tacl_a_00041. URL `https://doi.org/10.1162/tacl_a_00041`.

A. Borg, M. Boldt, O. Rosander, and J. Ahlstrand. E-mail classification with machine learning and word embeddings for improved customer support. *Neural Computing and Applications*, 33(6):1881–1902, 2021.

J. Brank, M. Grobelnik, N. Milic-Frayling, and D. Mladenic. Training text classifiers with svm on very few positive examples. Technical report, Technical Report MSR-TR-2003-34, Microsoft Corp, 2003.

C. Brun, D. N. Popa, and C. Roux. Xrce: Hybrid classification for aspect-based sentiment analysis. In *SemEval@ COLING*, pages 838–842. Citeseer, 2014.

N. Burns, Y. Bi, H. Wang, and T. Anderson. Sentiment analysis of customer reviews: Balanced versus unbalanced datasets. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 161–170. Springer, 2011.

G. Carenini, R. T. Ng, and E. Zwart. Extracting knowledge from evaluative text. In *Proceedings of the 3rd international conference on Knowledge capture*, pages 11–18, 2005.

G. Catanese, 2021.

J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408:189–215, 2020. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2019.10.118. URL `https://www.sciencedirect.com/science/article/pii/S0925231220307153`.

D. Chaturvedi and S. Chopra. Customers sentiment on banks. *International Journal of Computer Applications*, 98(13), 2014.

A. Chaudhary. A visual survey of data augmentation in nlp, 2020.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

C. Chu and R. Wang. A survey of domain adaptation for neural machine translation. *arXiv preprint arXiv:1806.00258*, 2018.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

D. R. Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.

M. Damaschk, T. Dönicke, and F. Lux. Multiclass text classification on unbalanced, sparse and noisy data. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 58–65, 2019.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

K. Doshi. Transformers explained visually (part 3): Multi-head attention, deep dive, Jan 2021. URL `https://towardsdatascience.com/transformers-explained-visually-part-3-multi-head-attention-deep-dive-1c1ff1024853`.

E. Fix and J. L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247, 1989.

R. Gandhi. Support vector machine - introduction to machine learning algorithms, Jul 2018. URL `https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47`.

X. Gu and S. Kim. " what parts of your apps are loved by users?"(t). In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 760–770. IEEE, 2015.

A. Gupta, V. Dengre, H. A. Kheruwala, and M. Shah. Comprehensive review of text-mining applications in finance. *Financial Innovation*, 6(1):1–25, 2020.

S. Ha, D. J. Marchetto, S. Dharur, and O. I. Asensio. Topic classification of electric vehicle consumer experiences with transformer-based deep learning. *Patterns*, 2(2): 100195, 2021.

M. A. Hadi and F. H. Fard. Evaluating pre-trained models for user feedback analysis in software engineering: A study on classification of app-reviews. *arXiv preprint arXiv:2104.05861*, 2021.

Z. S. Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, 2004.

S. Indra, L. Wikarsa, and R. Turang. Using logistic regression method to classify tweets into the selected topics. In *2016 international conference on advanced computer science and information systems (icacsis)*, pages 385–390. IEEE, 2016.

T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.

M. Jordan. Serial order: a parallel distributed processing approach. technical report, june 1985-march 1986. Technical report, California Univ., San Diego, La Jolla (USA). Inst. for Cognitive Science, 1986.

D. Jurafsky and C. Manning. Natural language processing. *Instructor*, 212(998):3482, 2012.

D. Jurafsky and J. H. Martin. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, 2009.

K. S. Kalyan, A. Rajasekharan, and S. Sangeetha. Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*, 2021.

J. Kazmaier and J. Van Vuuren. Sentiment analysis of unstructured customer feedback for a retail bank. *ORiON*, 36(1):35–71, 2020.

S. Khan. Bert, roberta, distilbert, xlnet-which one to use?, May 2021. URL `https://towardsdatascience.com/bert-roberta-distilbert-xlnet-which-one-to-use-3d5ab82ba5f8`.

S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 437–442, 2014.

G. J. Krishna, V. Ravi, B. V. Reddy, M. Zaheeruddin, H. Jaiswal, P. S. R. Teja, and R. Gavval. Sentiment classification of indian banks' customer complaints. In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pages 429–434. IEEE, 2019.

M. Kubat, S. Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*, volume 97, page 179. Citeseer, 1997.

R. Kusumawati, A. D'arofah, and P. Pramana. Comparison performance of naive bayes classifier and support vector machine algorithm for twitter's classification of tokopedia services. In *Journal of Physics: Conference Series*, volume 1320, page 012016. IOP Publishing, 2019.

J. D. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

B.-H. Leem and S.-W. Eum. Using text mining to measure mobile banking service quality. *Industrial Management & Data Systems*, 2021.

K. N. Lemon and P. C. Verhoef. Understanding customer experience throughout the customer journey. *Journal of marketing*, 80(6):69–96, 2016.

S. Li, Z. Wang, G. Zhou, and S. Y. M. Lee. Semi-supervised learning for imbalanced sentiment classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

B. Liu and L. Zhang. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer, 2012.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

S. Loukas. Text classification using naive bayes: Theory amp; a working example, Mar 2022. URL `https://bit.ly/3GG7l1O`.

W. Maalej, Z. Kurtanović, H. Nabil, and C. Stanik. On the automatic classification of app reviews. *Requirements Engineering*, 21(3):311–331, 2016.

A. Malte and P. Ratadiya. Evolution of transfer learning in natural language processing. *arXiv preprint arXiv:1910.07370*, 2019.

R. R. Mekala, A. Irfan, E. C. Groen, A. Porter, and M. Lindvall. Classifying user requirements from online feedback in small dataset environments using deep learning. In *2021 IEEE 29th International Requirements Engineering Conference (RE)*, pages 139–149, 2021. doi: 10.1109/RE51729.2021.00020.

T. Menner, W. Höpken, M. Fuchs, and M. Lexhagen. Topic detection: identifying relevant topics in tourism reviews. In *Information and communication technologies in tourism 2016*, pages 411–423. Springer, 2016.

G. Mihaila. fine-tune transformers in pytorch using transformers, Oct 2020. URL `https://gmihaila.medium.com/fine-tune-transformers-in-pytorch-using-transformers-57b40450635`.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

D. Mittal and S. R. Agrawal. Determining banking service attributes from online reviews: text mining and sentiment analysis. *International Journal of Bank Marketing*, 2022.

F. Mosteller and D. L. Wallace. Applied bayesian and classical inference : the case of the federalist papers. 1984.

Z. Mottaghinia, M.-R. Feizi-Derakhshi, L. Farzinvash, and P. Salehpour. A review of approaches for topic detection in twitter. *Journal of Experimental & Theoretical Artificial Intelligence*, 33(5):747–773, 2021.

A. Mountassir, H. Benbrahim, and I. Berrada. Addressing the problem of unbalanced data sets in sentiment analysis. In *International Conference on Knowledge Discovery and Information Retrieval*, volume 2, pages 306–311. SCITEPRESS, 2012.

J.-C. Na, H. Sui, C. Khoo, S. Chan, and Y. Zhou. Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews. *Advances in Knowledge Organization*, 9:49–54, 2004.

S. Pan and Q. Yang. A survey on transfer learning. ieee transaction on knowledge discovery and data engineering, 22 (10), 2010.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL `https://aclanthology.org/D14-1162`.

E. Potyraj. 4 ways to improve class imbalance for image data, Mar 2021. URL `https://towardsdatascience.com/4-ways-to-improve-class-imbalance-for-image-data-9adec8f390f1`.

R. C. Prati, G. E. Batista, and M. C. Monard. Class imbalances versus class overlapping: an analysis of a learning system behavior. In *Mexican international conference on artificial intelligence*, pages 312–321. Springer, 2004.

X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897, 2020.

I. Raicu. Financial banking dataset for supervised machine learning classification. *Informatica Economica*, 23(1), 2019.

A. Rajaraman and J. D. Ullman. *Data Mining*, page 1–17. Cambridge University Press, 2011. doi: 10.1017/CBO9781139058452.002.

A. Rantanen, J. Salminen, F. Ginter, and B. J. Jansen. Classifying online corporate reputation with machine learning: a study in the banking domain. *Internet Research*, 2019.

P. Ray and A. Chakrabarti. A mixed approach of deep learning method and rule-based method to improve aspect level sentiment analysis. *Applied Computing and Informatics*, 2020.

R. Rehurek and P. Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

I. Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf. Transfer learning in natural language processing. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials*, pages 15–18, 2019.

F. Rustam, M. Khalid, V. Rupapara, A. Mehmood, and G. S. Choi. A performance comparison of supervised machine learning models for covid-19 tweets sentiment analysis. *PLOS ONE*, 16:e0245909, 02 2021. doi: 10.1371/journal.pone.0245909.

J. Saias. Sentiue: Target and aspect based sentiment analysis in semeval-2015 task 12. Association for Computational Linguistics, 2015.

V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

M. Schuster and K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997. doi: 10.1109/78.650093.

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909, 2015a. URL `http://arxiv.org/abs/1508.07909`.

R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*, 2015b.

S. Singh and A. Mahmood. The nlp cookbook: Modern recipes for transformer based deep learning architectures. *IEEE Access*, 9:68675–68702, 2021.

I. Spasić, P. Burnap, M. Greenwood, and M. Arribas-Ayllon. A naïve bayes approach to classifying topics in suicide notes. *Biomedical informatics insights*, 5:BII–S8945, 2012.

C. Stanik, M. Haering, and W. Maalej. Classifying multilingual user feedback using traditional machine learning and deep learning. In *2019 IEEE 27th international requirements engineering conference workshops (REW)*, pages 220–226. IEEE, 2019.

C. Sun, L. Huang, and X. Qiu. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. *arXiv preprint arXiv:1903.09588*, 2019.

Z. Toh and J. Su. Nlangp: Supervised machine learning system for aspect category classification and opinion target extraction. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 496–501, 2015.

T. TowardAI. Sentiment analysis with logistic regression, Feb 2021. URL `https://towardsai.net/p/nlp/sentiment-analysis-with-logistic-regression`.

B. Van Aken, J. Risch, R. Krestel, and A. Löser. Challenges for toxic comment classification: An in-depth error analysis. *arXiv preprint arXiv:1809.07572*, 2018.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

F. Vencovskỳ, T. Bruckner, and L. Šperková. Customer feedback analysis: case of e-banking service. In *3rd European Conference on Social M di R h Media Research EM Normandie, Caen, France*, page 404, 2016.

E. Voita. Sequence to sequence (seq2seq) and attention, Apr 2022. URL `https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html`.

G. Wu and E. Y. Chang. Class-boundary alignment for imbalanced dataset learning. In *ICML 2003 workshop on learning from imbalanced data sets II, Washington, DC*, pages 49–56. Citeseer, 2003.

F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.