

## Comparative Analysis of Classical and Deep Learning-based Natural Language Processing for Prioritizing Customer Complaints

Jan Blümel  
University of Cambridge  
[jhb65@cam.ac.uk](mailto:jhb65@cam.ac.uk)

Mohamed Zaki  
University of Cambridge  
[mehyz2@cam.ac.uk](mailto:mehyz2@cam.ac.uk)

### Abstract

*Recent advancements in natural language processing have been shown to be very effective for different text mining tasks and thus have provided the opportunity to enhance service research. To improve the customer service experience, this paper compares several natural language processing approaches in order to automatically prioritize incoming customer complaints for service agents. This can help companies to reduce customers' friction and enable effective resource allocations. Our paper uses state-of-the-art feature engineering techniques (e.g., term frequency, TF-IDF and Word2Vec) to identify key words that could enable machine to prioritize complainers. We experimented with many classical machine learning classification algorithms, such as Random Forests, Support Vector Machines, Decision Trees and Logistic Regression, as well as with deep learning-based classifiers, such as convolutional neural networks, bidirectional long short-term memory, and the pre-trained language model BERT to compare the model performance. Our findings show that the pre-trained language model BERT and TF-IDF in combination with Logistic Regression yields the highest macro averaged F1-score across the multiple classes and is therefore most capable of predicting the priority group of incoming customer complaints.*

### 1. Introduction

Today, natural language processing (NLP) is one of the most promising subfields of artificial intelligence and in recent years has received considerable attention. NLP aims to read, understand, and derive meaningful insights from human languages. Because many customers communicate with a company in text form, NLP can help to gain

customer knowledge from sources within a company (e.g., contact centers) or outside a company (e.g., social media or online reviews) [1], [2]. Especially in service, many interactions have moved from offline to online platforms, increasing the availability of textual data [2]. The field of NLP has moved forward by a considerable margin and has unleashed new potential by the introduction of deep learning. Deep learning is known for its capabilities to learn complex patterns and is therefore an ideal complement for NLP tasks, as human language is often complex and ambiguous [3]. The superior performance for a variety of NLP tasks [4] can be seen in recent successes such as OpenAI's GPT-3 [5], which is able to write entire articles or computer codes [6]. Deep learning-based NLP can be of great use in the service domain and have already been applied in several use cases including customer satisfaction measurement with sentiment analysis [7], or self-service using AI-powered chatbots [8]. Further opportunities range from studying organizational frontlines using textual data to customer experience research [2].

To improve customer experience, it is necessary to reduce friction between the company and customer and foster a seamless customer experience [9]. Customer service acts as one major touchpoint between the customer and company, and is particularly important as it connects the company with the customer when they have a problem that needs to be fixed and affects them on many personal levels. By understanding the customer complaint, collecting feedback and delivering adequate customer service, both the company and the customer can benefit [10]. Within the last decade customer complaints have increased, because social media websites such as Twitter and Facebook have created an environment of instant feedback [10]. Dealing with the vast number of customer complaints requires effective processes and typically a lot of resources. If done insufficiently, it can leave customers waiting in queues or with unsolved issues.

Automatically prioritizing customer complaints using NLP can improve the customer service experience by reducing those waiting times for urgent matters. The prioritization and personalized treatment of incoming complaints can allow the company to allocate resources within customer service more effectively, as such content can rapidly overwhelm a company's information processing capabilities [11], [12]. Although some papers have looked at automatic prioritization of the customer voice, most focus on an identification of emerging topics through the analysis of past customer feedback only (see [13]–[15]). For example, review posts are prioritized based on frequency, sentiment, and sometimes timeliness. Only two papers address the aforementioned problem of effectively handling new incoming customer complaints [16], [17]. However both only used very limited NLP methods thus are missing out on the advancements of deep learning-based NLP.

Our research aims to address this gap by investigating the new advances in NLP for the automatic prioritization of incoming customer complaints. Therefore we conducted a comparative analysis of classical and deep learning-based NLP techniques to evaluate their suitability in analyzing and prioritizing incoming customer complaints. The contribution of this paper is two-fold. First it presents an easy to implement approach to prioritize incoming customer complaints based on historical data by using state-of-the-art NLP techniques. Therefore it shows the practical value of these NLP techniques for the service domain. The developed tool can help companies to automatically allocate resources, such as service agents more effectively, and helps to minimize friction on the customers' side, as more urgent problems are handled more efficiently. Second, it compares different NLP techniques applied to a small dataset to help practitioners in the decision-making process for deploying NLP in similar settings.

## 2. Background

Before we review existing approaches to prioritize customer complaints in chapter 2.2 and distil the research gap, chapter 2.1 presents basic theoretical concepts of NLP together with emerging topics. This shall give the reader an understanding and background of the tools used.

### 2.1. NLP pipeline

The application of NLP usually follows a simple process which starts with collecting the textual data and pre-processing it accordingly. Afterwards the text is represented by selected features during the feature

engineering process step, which is followed by a modelling technique.

Pre-processing the textual data is crucial for generating high value output [18]. It aims to prepare the textual data so that all non-informative information is excluded and a focus on the core information is possible. Commonly used techniques include punctuation, stop-word, or any kind of non-informative character removal. Additionally the text is sometimes stemmed, reduced to the stem of the word, or lemmatized, so that inflected forms of a words are grouped [1]. The textual data is then tokenized.

Feature engineering first extracts relevant features from the text that are able to represent it and then converts the textual data into numerical values so that they can be processed by a modelling algorithm. Classical techniques, such as term frequency (TF) or term frequency–inverse document frequency (TF-IDF), are mostly using the frequency of a single word (unigram) or an adjacent phrase (bi- or trigram) to represent the text. These are easy to implement techniques but lack a contextual understanding of a text, since they only capture the meaning of the extracted word itself. Furthermore, the techniques are hindered by the curse of dimensionality [4]. Word embeddings overcome this shortcoming to an extent, as they represent words by vectorization and thereby create a dense vector space [19]. Each word is represented by a vector, such that words with a similar meaning are closer together. These word embeddings can either be trained on a local dataset or pre-trained, having been taught on very large corpuses such as Wikipedia. Embeddings represent a paradigm shift in textual feature engineering. However they also lack contextual understanding, and are unable to identify idioms, phrases or the meaning of similar words given a specific context [4]. Contextual word embeddings were introduced within the last couple of years and use parallel attention mechanisms (transformers) to capture the contextual meaning of a word by giving these vector representations depending on the context [20]. These are trained on extraordinary large datasets and possess millions of trainable variables. Prominent examples include GPT-2&3, BERT or ELMo [21].

The last step includes the learning part for a particular task, allowing to learn patterns from the representation of textual data and perform the desired modelling. Alongside classical machine learning modelling techniques, deep learning techniques have been able to generate state-of-the-art results for many NLP tasks [4]. In general deep learning uses multiple layers of nodes (networks) to learn the representation of data with multiple levels of abstraction [22]. Domains such as speech recognition, vision, or NLP benefit from the superior capabilities of deep learning

compared to classical machine learning [4]. They have shown greater flexibility with the structure and format of input allowing the processing of unstructured data without extensive feature engineering [4]. Recurrent neural networks (RNNs) are suitable, especially for processing textual data, because they are capable of modelling sequential data and thereby consider previous words in a sentence [4]. Additionally, deep learning-based techniques scale better with large datasets and show more potential for transfer learning due to their capability of pre-training [23]. However, in contrast to classical machine learning, deep learning-based techniques usually require far more training data to achieve similar results and are also more costly in terms of resources and time.

## 2.2. Prioritizing customer complaints

We conducted a literature review to identify current approaches of handling customer complaints more effectively. The database Web of Science was searched using the terms “customer review” or “customer complaint” in combination with “text mining” or “natural language processing”. Within the identified literature, we focused on publications aiming to prioritize complaints or feedback (see Table 1). In the literature review in general, we found that whilst the customer voice is studied extensively, it is mostly done using public feedback databases. These are often the subject of research and are used to mine the customer opinion because they are publicly accessible. Popular topics addressed in these papers include opinion, sentiment and emotion mining. For these topics, the latest NLP techniques are often applied. For example, one paper uses deep learning-based embeddings such as Word2Vec, FastText, and GloVe, as well as the CNN and bi-LSTM architectures to detect emotions [24]. Another paper compares the performance of contextualized embeddings such as BERT with other deep learning-based architectures such as CNN and bi-LSTM in combination with GloVe for sentiment analysis of customer feedback [25].

Although these advanced NLP approaches have helped significantly to improve the performance and accuracy of sentiment analysis and opinion mining, they have not yet been used to prioritize incoming customer complaints. The overview in Table 1 shows that none of the publications use advanced deep learning-based NLP techniques.

Very few approaches have dealt with the automated prioritization of customer feedback and complaints. On one hand side, there are approaches to prioritize public customer feedback, which use NLP to automatically determine the most urgent improvements in app development based on customer

feedback [13]–[15], [26]. On the other side, some publications analyze customer complaints and their topics to prioritize action of the government, but not the complaint itself [27], [28]. Only two approaches address the automated prioritization of incoming customer complaints [16], [17]. One prioritizes complaints based on RFM parameters, frequency of previous issues, recency and executive response means. Also here, no use is made of the latest NLP techniques [16]. In the other approach, prioritization is based on the intensity of the sentiment and, in case the same intensity occurs in two complaints, on the first come-first served principle. Here, pre-processing steps such as tokenization, POS tagging, named entity extraction and support vector machines are used for sentiment analysis [17]. Thus, a research gap can be identified, which is particularly characterized by the lack of the latest NLP techniques for automated prioritization of incoming complaints. None of the approaches explore the benefits for simultaneously reducing customer friction and allocating company resources more efficiently in the case of automated prioritization. Therefore, the research question we address is: *How can advanced NLP support the efficient handling of customer complaints by automated prioritization in customer service?*

## 3. Methodology

To answer the research question, we used the Text Mining Analysis Roadmap [29]. The roadmap is based on the Cross-Industry Standard Process for Data Mining [30], which was adapted for service research and consists of six steps. The first step is to gain an understanding of the background. Thus, we conducted a literature review of relevant topics as described in chapter 2 and created an understanding of NLP pipelines. As the next step, we provide an overview of the case study and the underlying data in chapter 3.1 to understand the business and then the data. Since this also includes the preparation of the data, we outline the labelling process in chapter 3.2. Then, as a fourth step in chapter 3.3, we present a comparative analysis of NLP pipelines based on the dataset, examining how advanced NLP can help prioritize incoming customer complaints. In doing so, we address the individual steps of the NLP pipeline to design a suitable classification algorithm. As the fifth and sixth step, the results are evaluated in chapter 4 and the insights are discussed as a final step in chapter 5.

### 3.1. Case study background

We worked with a small to medium-sized German IT company to evaluate their customer complaints.

**Table 1: Literature overview of approaches prioritizing customer complaints**

<b>Paper</b>	<b>Analysis/ Purpose</b>	<b>Domain</b>	<b>Data Source</b>	<b>Technique</b>	<b>Prioritization Rule</b>
[31]	Prioritization of patient complaint and grievances for improving experience	Healthcare	9233 complaints in one hospital over a year	Excel tool with self-developed criteria	Classification in five priority groups based on a qualitative scale
[13]	Identification of key topics in user reviews to prioritize troubleshooting	IT/ App Developers	4,193,549 user reviews of 623 apps from Google Play Store	NLP pre-processing: Stop word and punctuation removal, stemming LDA for topic modelling	Key topics, which share significant relation to star-rating served as prioritization tool
[14]	Prioritization of user reviews for the purpose of app evolution	IT/ App Developers	725 user reviews from Google Play for 14 apps	NLP pre-processing: Stop word removal, stemming, n-gram extraction Random Forest for classification	Categories are ranked based on cardinality, oldest date, average rating, and importance of category evaluated by expert
[16]	Prioritization of the urban needs and estimate citizens' satisfaction based on citizens' complaint mining	Society	Database of '137 centers' in Bojnourd municipality with about 1500 entries	Impartial and repetitive data were deleted Frequency was calculated	Prioritization based on RFM parameters: recency, frequency, and executive response means
[15]	Prioritization of warnings, while developing an app by leveraging app user reviews	IT/ App Developers	About 26000 user reviews on six large-scale open-source Android apps	NLP pre-processing: Stop word removal, stemming, tokenization Feature Engineering: TF-IDF	Prioritization is based on similarity between the warning document and the user review
[32]	New process and decision support system for automotive defect identification and prioritization	Automotive	Discussion forums Honda-Tech.com, ToyotaNation.com, and ChevroletForum.com with 1500 threads	NLP pre-processing: Word sense disambiguation, word categorization, stemming, uni-gram Keyword based sentiment analysis Human expert tagging Logistic Regression for multi-class classification	Prioritization was done by differentiating between safety (urgent) and normal defects, which were identified based on pre-defined topics
[33]	Prioritization of different service quality parameter	Utility Service	Survey with 75 employees and 50 consumers	RIDIT analysis to determine service priority index for predetermined parameters	Calculated service priority index
[28]	Identification of urban residents wants regarding safety and disaster management from governments for prioritization of actions	Society	Social survey with civil complaint dataset of telephone-based complaints	NLP pre-processing: Tokenization, number removal Dictionary based word extraction Manual non-common noun extraction	Frequency-based prioritization
[17]	Prioritization of citizens' complaints based on sentiment	Society	<i>Not given</i>	NLP pre-processing: Tokenization, POS tagging, Named Entity Extraction Feature Engineering with WordNet SVM for Sentiment Analysis	Prioritization of higher intensity, followed by first come, and threshold time
[27]	Classification of citizens' complaints and proposals into topics	Society	56,708 complaints and proposals from community association and citizens of Jakarta through the e-Musrenbang system	NLP pre-processing: Tokenization, stemming, spell correction, lowercase, URL removal Feature Engineering: TF-IDF Classification with SVM	Prioritization based on topic frequency
[26]	Prioritization of feature improvements for app development	IT/ App Developers	4,442 reviews for the MyTracks app from the Google Play Store	NLP pre-processing: Sentence parser, character removal, filter based on low ratings, noun extraction with POS tagging and n-gram Emotion detection with LIWC dictionary	Prioritization based on frequency of terms, rating, negative emotions, and deontics

The company offers IT services, such as IT infrastructure and security for business customers. The company received 520 customer complaints in the period from March 2020 to January 2021, which were previously handled on a first-come first-served basis. Customer complaints were stored as free-text messages in a ticketing system in an SQL database. The data submitted included a text field, date, customer id and ticket number. To prioritize the incoming customer complaint, we focused on the textual data. Thus we extracted the textual information in json format from the SQL database and transformed it into a python Dataframe for further analysis. The 520 extracted customer complaints had an average length of about 37 words with some very long outlier complaints of 677 words. However, such long complaints are very rare, as the mode of the complaints is 10 words.

### 3.2. Data labelling

The dataset we obtained was unlabeled because the company did not prioritize their customer complaints. Since other information such as ratings of complaints were not available as in previous studies [14], [26], the prioritization was done completely based on the text conveyed.

To prioritize the existing customer complaints, we assumed that the complaints collected over approximately one year are representative. Furthermore we assumed that in general, as based in best-practice management tools such as ABC analysis or the pareto principle, the worst and most urgent 20% of complaints have a very large impact on the customer and the company, and therefore need to be resolved most quickly. The next 30% of the most urgent complaints must then be dealt with, as these have a medium impact on the customer and the company. The remaining 50% of complaints have the lowest priority. This is an approximation based on the objective to effectively allocate existing resources within the company. The task of the labelling process was then to find the complaints in the dataset that belong to the 20% of the most urgent complaints (group with priority 1), the following 30% of complaints (group with priority 2) and the remaining complaints (group with priority 3). This ensures a high impact approach by which resources in customer service could ultimately be efficiently allocated. To find these complaints, we used three different decision variables that assign a value to each complaint. The sum of these values then formed the score of each complaint, and then all complaints were ranked on this new score. The top 20% of complaints with the highest score were then assigned to the highest priority group

1, the next 30% to priority group 2 and the 50% of complaints with the lowest score are assigned to priority group 3.

**Decision variable 1:** We argue that, in general, the subjectivity of a complaint is an indication of how urgent the complaint is. Simplified, the more subjective the language, the higher the likelihood that the customer needs help urgently. Similar arguments have been made by other approaches that use sentiment analysis to determine the priority of a complaint or feedback [26], [34]. The subjectivity of a complaint was calculated using the lexicon-based German Text-Blob API, a python library, which calculates a value between 0 and 1 for a complaint, where 1 is strongly subjective.

**Decision variable 2:** We also argue, based on sentiment-oriented prioritization, that the polarity of a complaint is an indication of its urgency (see [26]). The polarity reflects the customers' attitude and dissatisfaction. Because these dissatisfied customers with higher polarity in their language are more likely to churn than other customers, their complaints should be handled with a higher urgency to avoid a customer churn. The polarity was calculated using the lexicon-based German Text-Blob API, which assigns a value to a complaint ranging from -1 for very negative to 1 for very positive.

**Decision variable 3:** For the third decision variable, we followed similar approaches from chapter 2.2, which prioritized feedback or complaints using the frequency of topics within the customer complaints and feedback [16], [26]–[28]. We argue that the more frequently a topic appears in complaints, the more urgent it is to solve it. Therefore a complaint on a highly frequent topic should be prioritized. To identify the frequent topics in the complaints, we extracted representative keywords/-phrases from the complaints. We first tokenized the complaints into uni-, bi-, and trigrams and removed stop words. We then used the contextualized word embedding, provided by the German language model BERT [35], pretrained on Wikipedia, news and a total of 12GB of data, to vectorize the n-grams. The advantage of using BERT is that it is already pretrained therefore it knows many different words and their context to each other. Based on the determined embeddings, the cosine-similarity between the keywords/-phrases and the complaints was calculated, with the assumption that the more similar the words and phrases are to the complaints, the more likely they are to represent them. After determining the representative keywords/-phrases, the top 100 most frequently occurring ones were examined for their occurrence in the individual complaints. Each complaint was given a frequency value. Whether the keywords/-phrases were

representative was cross-checked by random sampling. 10% of the complaints were checked against their top 3 keywords/-phrases by an employee of the company and the author.

After calculating three values for each complaint, we normalized each of these across all complaints and calculated a score per complaint, adding decision variable 1 and 3 and then subtracting decision variable 2. The division into the three categories was then done as described above. As a result, we obtained a labelled dataset.

### 3.3. NLP pipeline applied

The following section explains how we built a classifier, which learnt how to automatically prioritize incoming customer complaints. We present the individual steps of the NLP pipeline and the findings of a comparative analysis among them. The environment used was Google Colab with a CPU model Intel(R) Xeon(R) CPU @ 2.30GHz and 12GB RAM size. Google Colab's GPU resources (Nvidia K80 @ 1.59GHz und 12GB Memory) were used for hyperparameter training but not for the final classification evaluation.

**3.3.1. Pre-processing.** As the raw customer complaints were conveyed with contextual data such as e-mail addresses, or links, pre-processing steps needed to be performed in order to extract the written complaint into a usable format. However even plain customer complaint texts sometimes needed to be pre-processed, because the raw textual data is full of non-informative words and characters [18]. The pre-processing steps performed were thereby dependent on the feature engineering technique used. The classical, frequency-based feature engineering techniques TF or TF-IDF are sensitive to noise and repeating non-informative words such as 'and' or 'not'. As a result, we conducted tokenization, stop word and punctuation removal, stemming and lemmatization (Type A). We also investigated the pretrained Word2Vec embedding, which during the pretraining phase conducted pre-processing as well [36]. Therefore we adapted the same steps as performed during the pretraining phase for our application and pre-processed the data by performing only tokenization, stop word, punctuation removal, and umlaut replacement (e.g. ö to oe) (Type B).

**3.3.2. Feature engineering.** We applied TF and TF-IDF to our dataset, as the most commonly used classical NLP techniques for feature engineering. The python sklearn feature extraction package with the Count and TF-IDF vectorizers were used in

combination with n-grams (uni-, bi- and trigrams). For comparison, we also used a pre-trained deep learning-based German Word2Vec embedding using skip-gram and CBOW models for feature engineering. The model was pretrained on German news and Wikipedia corpus with a vocabulary of about 600k words [36].

**3.3.3. Modelling.** As explained in chapter 3.2, three different classes were used to prioritize incoming customer complaints. To learn the classification of the complaints, we investigated a variety of popular classical machine learning techniques such as Support Vector Machines, Decision Trees, Logistic Regression, and Random Forests. We performed hyperparameter tuning on these models to allow a comparative analysis on the individually best setting. Therefore we performed a grid search with 5-fold cross validation on the training dataset, scored on the F1 macro. The best models which were used in the comparative study can be found in Table 2. The two popular architectures for deep learning-based classifiers were also investigated: a convolutional neural net (CNN) and a bidirectional long short-term memory (bi-LSTM). The bi-LSTM was chosen because it has shown better performance than other RNNs as it overcomes the vanishing and exploding gradient problem [4]. On both architectures limited hyperparameter tuning was performed over the batch size and number of epochs, using the same cross-validation technique as before. The hyperparameters chosen are displayed in Table 2. The architecture of the CNN model was built out of 6 layers: an embedding layer, a convolutional 1D layer, followed by a max pooling and a flatten layer and then two dense layers, the last of which was used for the final classification. The bi-LSTM model was built out of the embedding layer, a spatial dropout and the bidirectional LSTM layer, followed by the dense layer with output dimension 3. Lastly, we explored a pre-trained BERT model for classifying incoming customer complaints. The transformer-based model represents the latest advancements in NLP and makes use of attention mechanisms to capture context in textual data. We used a pre-trained Bert multilingual base model that was pretrained on 102 languages, including German, on Wikipedia [37]. We used this model, because it has a sequence classification setting which can be used for multi-class classification.

For the training in general, we split the dataset into a test dataset and a training dataset on which the training and cross-validation was performed. The split of training to test dataset was 70:30 to ensure enough representation of all classes in the test set. Since the dataset was, by nature, slightly imbalanced, appropriate measures had to be taken so that the

algorithms were not trained on the majority class. Otherwise, the accuracy of the classification model would have been high, but it would only have predicted low priority customer complaints. Thus, we trained the dataset on class weights, assigning every

class a weight, so that all categories are balanced. To control for overfitting in training the deep learning-based models, we included an early stop function which stopped the training if the performance on the validation set didn't improve.

**Table 2: NLP pipelines applied to customer complaint prioritization**

#	Pre-processing	Feature Engineering	Model (Hyperparameters)
1	Type A	TF	Random Forest (Criterion = gini, n_estimators = 100)
2	Type A	TF	Logistic Regression (C=100, Penalty = l2, Solver = liblinear)
3	Type A	TF	Support Vector Machine. (C=0.1, Kernel = linear, Gamma = scale)
4	Type A	TF	Decision Tree (Criterion = gini, Splitter = best)
5	Type A	TF-IDF	Random Forest (Criterion = gini, n_estimators = 10)
6	Type A	TF-IDF	Logistic Regression (C=1000, Penalty = l2, Solver = saga)
7	Type A	TF-IDF	Support Vector Machine (C=10, Kernel = sigmoid, Gamma = scale)
8	Type A	TF-IDF	Decision Tree (Criterion = gini, Splitter = random)
9	Type B	Pre-trained Word2Vec	Logistic Regression (C=1000, Penalty = l2, Solver = liblinear)
10	Type B	Pre-trained Word2Vec	Support Vector Machine (C=1000, Kernel = rbf, Gamma = scale)
11	Type B	Pre-trained Word2Vec	Decision Tree (Criterion = gini, Splitter = random)
12	Type B	Pre-trained Word2Vec	Random Forest (Criterion = entropy, n_estimators = 10)
13	Type A	TF	CNN (Layers: 6, Epochs: 15, Batch size: 32, Optimizer: adam, Learning rate: 0.001, Dropout rate: 0.0)
14	Type A	TF	Bi-LSTM (Layers: 4, Epochs: 10, Batch size: 32, Optimizer: adam, Learning rate: 0.001, Dropout rate: 0.2)
15	Type B	Pre-trained Word2Vec	CNN (Layers: 6, Epochs: 15, Batch size: 32, Optimizer: adam, Learning rate: 0.001, Dropout rate: 0.0)
16	Type B	Pre-trained Word2Vec	Bi-LSTM (Layers: 4, Epochs: 10, Batch size: 32, Optimizer: adam, Learning rate: 0.001, Dropout rate: 0.2)
17	BERT multilingual base tokenizer (uncased) with max_length = 100		BERT multilingual base model uncased (Epochs: 10, Optimizer: AdamW, Learning rate: 1e-5)

## 4. Results

The 17 NLP pipelines were evaluated based on their capability to predict the correct priority class of an incoming customer complaint. We use the F1 score over the accuracy measure as a performance indicator, because our dataset is imbalanced. The F1 score is the harmonic mean of the precision and recall. To take into account the classification of all classes and not to overestimate the importance of the majority class, we used the macro average F1 score as performance measure. The results can be seen in Table 3, with deep learning-based approaches highlighted in bold. In addition, we compared the computational time of the different approaches and their memory usage to evaluate the algorithmic efficiency.

The assumption that contextual embeddings based on transformer models perform the best because

they can account for the context of a word, was partially proven. We can see that a pre-trained BERT model, trained on a large corpus, outperforms almost all other approaches, with the exception of the Logistic Regression which, together with TF-IDF, performs equally well. However the margin between the macro F1 scores is less than expected and compared to the majority of approaches the large pretraining and fine-tuning time is not reflected in a higher macro average F1 score. This may be due to the different settings and datasets in pretraining and fine-tuning. It is worth noting here that most transformers are currently trained in English not in German.

Our results do not reflect the fact that the pre-trained Word2Vec model tends to perform better than TF or TF-IDF. Word2Vec generally allows the capture of more context and meaning while pretraining on a large corpus. The average macro average F1 score is

highest for TF-IDF, followed by TF and then Word2Vec. In comparison with TF and TF-IDF, Word2Vec is never the best feature engineering technique for a fixed modelling approach. Despite extensive pre-training of the Word2Vec embedding in German, we can observe that TF and TF-IDF yield at least comparable results.

By comparing the classical machine learning models with the deep learning-based models (which are known to require large datasets) we can observe that they all perform on average better than the deep learning-based models. In general, Logistic Regression outperforms Random Forests, bi-LSTM, and Decision Trees for all feature engineering techniques, whilst Support Vector Machines outperform bi-LSTM. We mainly attribute this to the fact that deep learning-based models require enough data to perform adequately, which is not the case with our small dataset. Therefore we can see that classical machine learning models are well suited compared to deep-learning models on this classification problem. Pipeline 15 has the highest F1-score for the priority 1 class. This is important because one objective of the prioritization is to reduce the friction of customers with the most urgent complaints.

The analysis shows that although deep learning-based techniques are promising, they don't always have the capability to perform equally well as classical machine learning pipelines, given the relatively small dataset. However classical machine learning NLP pipelines still yield comparable results and Logistic Regression, in combination with TF-IDF, have the highest average F1-score. This demonstrates the usability of the NLP techniques for customer complaint prioritization. The pre-trained language model BERT shows the capabilities of pretraining and of the transformer architecture as it too yields the highest average F1 score.

The computational time for training the NLP pipeline varies a lot across the different approaches once the training of the model and computation of the embedding on the training dataset have been taken into account. Hyperparameter training and pretraining are not included. Deep learning-based classification modelling techniques have significantly higher computational times. CNN, bi-LSTM and particularly the fine-tuning stage of BERT stand out by needing magnitudes of more training time when using the same feature engineering steps. This shows that the high training time of BERT (as well as CNN and bi-LSTM) is due to the fact that they were designed to be trained on GPUs instead of CPUs, as is the case here. Using pre-trained Word2Vec embeddings for classical machine learning does not considerably increase the computational time. However, for deep learning-based

classification algorithms using Word2Vec as a first layer does increase the computational time.

The significant differences among the approaches highlight the importance of considering this variable in choosing the appropriate approach. Despite yielding similar accuracies, a higher computational time can cause higher costs.

**Table 3: Performance matrix of NLP pipelines for customer complaint classification**

#	F1-score				Time [s]	Memory [Mb]
	1	2	3	Macro avg.		
1	0.42	0.58	0.75	0.58	0.54	0.5
2	0.55	0.57	0.74	0.62	0.07	0.5
3	0.53	0.64	0.79	0.65	0.06	0.5
4	0.63	0.52	0.71	0.62	0.07	0.5
5	0.52	0.61	0.74	0.62	0.09	0.5
6	0.62	0.62	0.76	0.67	0.12	0.5
7	0.50	0.59	0.74	0.61	0.07	0.5
8	0.53	0.59	0.72	0.61	0.1	0.5
9	0.56	0.61	0.76	0.64	0.19	10.8
10	0.60	0.59	0.74	0.64	0.18	10.8
11	0.49	0.58	0.72	0.60	0.09	10.8
12	0.48	0.59	0.68	0.58	0.17	10.8
13	0.65	0.54	0.76	0.65	42.71	1.0
14	0.62	0.58	0.59	0.60	209.92	1.3
15	0.67	0.51	0.57	0.58	326.83	7.5
16	0.56	0.45	0.67	0.56	1259.5	9.6
17	0.62	0.58	0.82	0.67	1h14m	40.6

The memory usage of the algorithm does not deviate as much as the training time, with exception of the BERT model. However we can still observe considerable differences between classical approaches using 0.5Mb, compared to Word2Vec based approaches which need between 7.5 and 10.8Mb. As with the computational time, this variable can be used for decisions relating to similar accuracies as it influences the costs associated with the project.

## 5. Discussion

First, we present a method rooted in the literature to prioritize customer complaints based on heuristics. Established NLP methods for the identification of polarity and subjectivity, as well as the latest transformer models for the identification of keywords/-phrases, were used to follow these heuristics and automatically label historical complaints. This method has the advantage that it can be implemented quickly and without extensive expert or domain knowledge. Depending on the language the models are pre-trained on, adaptation to other

languages and industries is easily possible. However, especially when using the pre-trained transformer, care must be taken not to have too much out-of-vocabulary text modules in the complaints, for example due to specialized domains. Due to the automatic labelling technique using a keyword- and sentiment-based approach, the customer complaint classification may exhibit slightly inaccurately labelled data. The customer-centric approach could have benefited from using customer judgement to increase the representative power of the labels.

Secondly, the results of the comparative analysis showed that it is possible to classify newly arriving customer complaints relatively accurately into a priority group based on the historical data, even for small data sets. This is especially helpful for smaller companies that do not have access to large domain-specific datasets. If successfully labeled training data sets are already available, the classical methods have the advantage that no Word2Vec or transformer models, pre-trained on large data sets, must be present. Therefore an adaptation into other languages is simpler. In addition, the mostly low computation time and memory usage show that the method can be implemented in a resource-efficient way.

Moreover, this research addresses the academic gap by combining the need to prioritize incoming customer complaints (to allocate resources more effectively) with several advanced NLP methods, such as Word2Vec and transformers (which are designed to better understand the context of texts). This research goes beyond existing research in that it not only analyzes historical complaints to identify important issues, as in our labelling process, but also presents an application for everyday use to prioritize newly incoming complaints.

In this paper we have used three different deep learning-based classifiers and two deep learning-based feature engineering techniques and compared them with classical approaches. Future research in this area could extend the comparison by investigating further deep learning-based techniques such as GloVe [38] for feature engineering or GRUs and other transformer models such as XLM or GPT-3 for classifications. Future research could also investigate the dependency of deep learning-based pipelines on the dataset size by analyzing larger datasets. Additionally, implementation in practice could help to justify the value of NLP in customer complaint prioritization.

## 6. Conclusion

This paper presents a comparative analysis of NLP techniques for prioritizing incoming customer complaints. It has shown that classical and deep

learning-based NLP pipelines achieve comparable F1-scores, although deep learning-based approaches require higher magnitudes of computational time and memory. Using the pre-trained BERT model for sequence classification and TF-IDF with Logistic Regression achieves the highest macro average F1-score of 67%. The results demonstrate a fast and computationally light approach to customer complaint prioritization in small datasets, using classical and deep learning-based NLP.

## 7. References

- [1] J. Berger, A. Humphreys, S. Ludwig, W. W. Moe, O. Netzer, and D. A. Schweidel, "Uniting the Tribes: Using Text for Marketing Insight," *J. Mark.*, vol. 84, no. 1, pp. 1–25, Jan. 2020,
- [2] F. Villarreal Ordenes and S. Zhang, "From words to pixels: text and image mining methods for service research," *J. Serv. Manag.*, vol. 30, no. 5, pp. 593–620, Nov. 2019,
- [3] M. M. Lopez and J. Kalita, "Deep Learning applied to NLP," *arXiv*, Mar. 2017, Accessed: May 18, 2021. [Online]. Available: <http://arxiv.org/abs/1703.03091>
- [4] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing [Review Article]," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018,
- [5] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," *arXiv*, May 2020, Accessed: May 18, 2021. [Online]. Available: <http://arxiv.org/abs/2005.14165>
- [6] R. Sagar, "OpenAI Releases GPT-3, The Largest Model So Far," Jun. 03, 2020. <https://analyticsindiamag.com/open-ai-gpt-3-language-model/> (accessed May 17, 2021).
- [7] Q. T. Ain *et al.*, "Sentiment Analysis Using Deep Learning Techniques: A Review," 2017. Accessed: May 16, 2021. [Online]. Available: [www.ijacsa.thesai.org](http://www.ijacsa.thesai.org)
- [8] X. Chi, H. Wang, Z. Wang, S. Chen, and X. Xu, "Predicting the Evolution of Service Value Features from User Reviews for Continuous Service Improvement," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10601 LNCS, 2017, pp. 142–157.
- [9] K. N. Lemon and P. C. Verhoef, "Understanding Customer Experience Throughout the Customer Journey," *J. Mark.*, vol. 80, no. 6, pp. 69–96, Nov. 2016,
- [10] V. Singh, A. Jain, and S. Choraria, "Exploring the Role of Complaint Handling among Complaining Consumers," *Vision*, vol. 20, no. 4, pp. 331–344, Dec. 2016,
- [11] C. Homburg, M. Droll, and D. Totzek, "Customer

- prioritization: Does it pay off, and how should it be implemented?," *J. Mark.*, vol. 72, no. 5, pp. 110–130, Sep. 2008,
- [12] M. Parameswaran and A. Whinston, "Research Issues in Social computing," *J. Assoc. Inf. Syst.*, vol. 8, no. 6, pp. 336–350, Jun. 2007,
- [13] E. Noei, F. Zhang, and Y. Zou, "Too Many User-Reviews! What Should App Developers Look at First?," *IEEE Trans. Softw. Eng.*, vol. 47, no. 2, pp. 367–378, Feb. 2021,
- [14] L. Etaiwi, S. Hamel, Y.-G. Gueheneuc, W. Flageol, and R. Morales, "Order in Chaos: Prioritizing Mobile App Reviews using Consensus Algorithms," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, Jul. 2020, pp. 912–920.
- [15] L. Wei, Y. Liu, and S.-C. Cheung, "OASIS: prioritizing static analysis warnings for Android apps based on app user reviews," in *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, Aug. 2017, vol. 11, pp. 672–682.
- [16] Ghodousi, Alesheikh, Saeidian, Pradhan, and Lee, "Evaluating Citizen Satisfaction and Prioritizing Their Needs Based on Citizens' Complaint Data," *Sustainability*, vol. 11, no. 17, p. 4595, Aug. 2019,
- [17] K. V. Deshmukh and S. S. Shiravale, "Priority Based Sentiment Analysis for Quick Response to Citizen Complaints," in *2018 3rd International Conference for Convergence in Technology (I2CT)*, Apr. 2018, pp. 1–5.
- [18] C. Knoblock, D. Lopresti, S. Roy, and V. L. Subramaniam, "Special issue on noisy text analytics," *International Journal on Document Analysis and Recognition*, vol. 10, no. 3–4. Springer, pp. 127–128, Dec. 20, 2007.
- [19] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, 2003.
- [20] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, Jun. 2017, vol. 2017-December, pp. 5999–6009. Accessed: May 11, 2021. [Online]. Available: <https://arxiv.org/abs/1706.03762v5>
- [21] M. E. Peters *et al.*, "Deep contextualized word representations," in *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, Feb. 2018, vol. 1, pp. 2227–2237.
- [22] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015.
- [23] S. Ruder, M. Peters, S. Swayamdipta, and T. Wolf, "Transfer Learning in Natural Language Processing," *NAACL HLT 2019 - Tutorial*, 2019.
- [24] E. Batbaatar, M. Li, and K. H. Ryu, "Semantic-Emotion Neural Network for Emotion Recognition From Text," *IEEE Access*, vol. 7, pp. 111866–111878, 2019,
- [25] A. Benlahbib and E. H. Nfaoui, "Aggregating customer review attributes for online reputation generation," *IEEE Access*, vol. 8, pp. 96550–96564, 2020,
- [26] S. Keertipati, B. T. R. Savarimuthu, and S. A. Licorish, "Approaches for prioritizing feature improvements extracted from app reviews," in *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, Jun. 2016, vol. 01-03-June, pp. 1–6.
- [27] I. B. N. Sanditya Hardaya, A. Dhini, and I. Surjandari, "Application of text mining for classification of community complaints and proposals," in *2017 3rd International Conference on Science in Information Technology (ICSITech)*, Oct. 2017, vol. 2018-Janua, pp. 144–149.
- [28] E. Lee, S. Lee, K. S. Kim, V. H. Pham, and J. Sul, "Analysis of Public Complaints to Identify Priority Policy Areas: Evidence from a Satellite City around Seoul," *Sustainability*, vol. 11, no. 21, p. 6140, Nov. 2019,
- [29] M. Zaki and J. R. McColl-Kennedy, "Text mining analysis roadmap (TMAR) for service research," *J. Serv. Mark.*, vol. 34, no. 1, pp. 30–47, 2020,
- [30] P. Chapman *et al.*, "CRISP-DM 1.0: Step-by-step data mining guide," 2000.
- [31] S. Bayer, P. Kuzmickas, A. Boissy, S. L. Rose, and M. B. Mercer, "Categorizing and Rating Patient Complaints: An Innovative Approach to Improve Patient Experience," *J. Patient Exp.*, vol. 8, p. 237437352199862, Jan. 2021,
- [32] A. S. Abrahams, J. Jiao, G. A. Wang, and W. Fan, "Vehicle defect discovery from social media," *Decis. Support Syst.*, vol. 54, no. 1, pp. 87–97, Dec. 2012,
- [33] S. Chatterjee and A. Chatterjee, "Prioritization of Service Quality Parameters Based on Ordinal Responses," *Total Qual. Manag. Bus. Excell.*, vol. 16, no. 4, pp. 477–489, Jun. 2005,
- [34] J. Patel, R. Dubey, and R. kumar Gupta, "PMI-IR Based Sentiment Analysis Over Social Media Platform for Analysing Client Review," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 44, Springer, Cham, 2020, pp. 204–212.
- [35] B. Chan, T. Möller, M. Pietsch, and T. Soni, "German BERT," *Hugging Face*. <https://huggingface.co/bert-base-german-cased> (accessed May 18, 2021).
- [36] deepset, "Pretrained German Word Embeddings." <https://deepset.ai/german-word-embeddings> (accessed May 18, 2021).
- [37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Oct. 2018, [Online]. Available: <http://arxiv.org/abs/1810.04805>
- [38] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.