Intelligent Customer Pathway

submitted in partial fulfillment for the degree of master of science

Silvio Affolter
11845953

master information studies
data science
faculty of science
university of amsterdam

2018-06-28

	Internal Supervisor	External Supervisor
Name	Dr. Maarten Marx	Zubair Afzal
Affiliation	UvA, FNWI, IvI	Elsevier
Email	maartenmarx@uva.nl	m.afzal.1@elsevier.com

Intelligent Customer Pathway

Automation of customer service by classifying incidents automatically from email messages.

Silvio Affolter
University of Amsterdam
silvio.affolter@gmail.com

ABSTRACT

Assigning contact reasons to incoming customer service emails is a labor intensive and error prone task. Achieving human performance with an automated system can lead to increased customer satisfaction at lower cost. We used a training set containing 22,552 first email messages, received by Elsevier customer service to train support vector machines and recurrent neural networks for automated email classification. Each email message has one out of eight possible contact reasons assigned by customer service agents. We used Krippendorff's alpha to measure human performance on an independent and stratified evaluation set of 500 emails messages, which were manually labelled by two independent annotators. Cross-validation was used to train automated models and measure their performance. To compare the performance of automated methods with human performance, we built models using the full training set and tested on the evaluation set. The performance of several classifiers was similar. The results of the human performance measure revealed the difficulty of the task. The research has demonstrated a SVM or bidirectional LSTM can outperform or reach the current human performance at the task of classifying emails with a contact reason.

1 INTRODUCTION

Customer service departments are confronted every day with the processing of a large number of emails from customers. To handle the workload, organizational structures have been established, especially multi-tiered customer service structures. In the last decade, automating customer services using intelligent methods has been trending and is expected to be ubiquitous within a few years [12]. The goal of automation is to improve the customer service experience without increasing the operational cost [2]. Automated classification of customer service emails by contact reason is a multi-class classification problem. A large amount of research has already been performed on email classification and even more on text classification [19] [15]. In the specific domain of customer service, little research has been published. The related literature found by the author for classifying customer service messages based on their content does not involve human performance measures for the task. In most general text classification studies, human performance is not considered either. Hence, this research focuses on comparing automated customer service email classification performance with human performance.

At Elsevier the first tier of customer support is responsible for identifying high level contact reasons and solving them right away if the problem is within their scope of complexity. A tiered organizational structure aims to keep the workload as low as possible for the second and third tiers of support [21]. The second and third levels are specialized departments for specific non-standard issues. Recent research claims that a combination of human interaction and automation is the most beneficial setup for customer service success [3]. One of the main factors of customer satisfaction with the service provided is the time needed to resolve the problem [7]. Automation can speed up the progress by forwarding the email instantly to the right person without waiting in a general queue for processing. The most important factor for the success of an automated system is the performance. If it can be shown that an automated system reaches the human performance level, chances for acceptance are high.

Van den Poel and Coussement [2] use machine learning methods for classification of customer service email and identify cost savings. Research of Jeltey et al. [8] focuses on customer service analytics by using automated classification methods. Recent research for customer service is mainly focusing on fully automated service chat bots [23]. In all of the machine learning text classification studies found by the author human peformance is not assessed for the classfication task. Only one report [22] has been found by the author where human performance is measured on a standard multi-class classification datastet. Therefore, the focus of the study is to assess if human performance of customer service agents can be reached by a machine learning classifier.

In this study, the focus is solely on the first level of customer support at Elsevier, which is split into three different queues specialized in specific contact reasons. The algorithms are trained on a dataset of 23,052 first email messages sent from the customer to customer service, which are labelled with a contact reason. For each message one out of eight contact reasons needs to be assigned. Based on the assigned contact reason, one out of three queues can be derived to determine where the message is to be routed. Each queue has two or three fixed contact reasons to process. The messages have a median length of 122 words.

Research questions. The goal of the research is to determine whether automated text classification methods can reach the human performance level of customer service agents in the task of labelling customer emails with a contact reason by using only the text body. The research is split into four subquestions.

(1) What is the current human service agent performance and can the system dataset labels be validated by the observers?
(2) How accurately can the text body of the first message from the customer to the customer service be extracted?
(3) What performance level can be achieved with a baseline model by using a well-performing (according to literature) standard multi-class text classification pipeline setup?
(4) How much can the automated classification performance be improved by using different pre-processing steps, feature selection methods, sampling techniques for imbalanced datasets, hyperparameter tuning, and alternative models?

This document is structured into several parts. After the introduction, the related literature relevant to each of the research questions is introduced. Later, the dataset is described and the methods used to answer the research questions are explained. Furthermore, the evaluation part includes results of all the sub-questions of the research and at the end, a conclusion regarding the main research question is drawn from the results of the sub-questions.

2 RELATED WORK

2.1 Text classification

The classification of emails based on the text body is a specific task of text classification. Machine learning models replaced rule-based systems years ago in the area of text classification [19]. Hence, only machine learning models are considered in this study. The main focus is on text classification using support vector machine (SVM) classifiers. Additionally, due to the increasing popularity of neural network approaches [25], recurrent neural networks (RNNs) with long short-term memory (LSTM) cells are considered for improvements. The focus is on an SVM baseline because the training set is not large and because literature reports good performance of multi-class classification tasks using support vector classifiers [9] [14].

Manning et al. [14] provide an introduction to support vector classifiers and divide them into two main groups of SVM, depending on whether they have a linear or a nonlinear kernel. They state that an SVM model is not necessarily better than other models but can provide state-of-the-art performance for many small or medium-sized datasets. According to a literature review about text classification algorithms [1], SVMs are most frequently used with the linear kernel in practice due to the simplicity of the kernel and the stable results. Training an SVM requires choosing between a one-versus-all approach and a one-versus-one approach [14]. The research by Rifkin and Klatau [18] elaborates that, in many situations, a one-versus-all approach should be preferred. A different aspect of training SVMs is the text representation, which needs to be in a vectorized form. According to literature, the bag-of-words (BOW) model is mainly used in combination with SVM, where each word or n-gram is represented as a dimension in the vector space [19]. Another approach of representing the text input is by using techniques like latent semantic analysis, where language models are created with lower dimensional and less sparse vectors [14]. A comparison of the text representation methods is provided by Zhang et al. [26]. A further aspect is the term weighting, where the tf-idf method is a common choice [14]. In literature, a lot of strategies have been developed for improving multi-class text classifiers. An approach that can lead to better classification performance is the use of ensembles of classifiers. The voting and bagging approaches are described by Aggarwal [1]. Other strategies only tackle the imbalance, which is high in the dataset of this study, by oversampling and under-sampling to improve the overall performance. The study by Estabrooks et al. [4] shows that sampling can be effective on a standard dataset.

In the past few years, research in text classification methods has shifted toward methods that use artificial neural networks. The paper by Young et al. [25] provides an extensive overview of recent deep-learning-based natural-language-processing methods. Two major popular architecture types of neural networks are used for natural language processing tasks—the convolutional neural networks (CNN) and the RNNs, which gained popularity mainly through image recognition and were later adapted to text classification. A comparison of the two approaches supports the indication that situations where the full sentence needs to be understood are better suited to RNN than CNN [24]. This study only focuses on RNN because it is assumed that some of the classes are semantically close and hard to distinguish without understanding the full sentence. Two common feature-representation techniques are mainly used with RNNs—the BOW representation, which leads to long and sparse vectors, and denser short vectors learned from the text data [25]. Pretrained global vectors have shown good performance as embedding layer because they can be computed on a large corpus and represent semantics [17]. Bi-directional neural networks are a special type of RNN; they take the previous and the following steps into account to assess a decision. In combination with the recurrent network, LSTM cells are used to tackle the problem of vanishing or exploding gradients while training RNNs [5]. Studies show that a bi-directional LSTM can outperform an SVM classifier on a standard multi-class dataset like the 20newsgroups dataset [13].

2.2 Performance measurement

2.2.1 Human labelling performance for emails. Measuring the human performance for a text classification task is important as all classes are not always easily identifiable. There is often some degree of ambiguity in free text—multiple people classifying the same text do not agree regarding the label. Measures have been developed to interpret results obtained by different people on the same text. The most complete introduction to content analysis and measuring agreement between observers is provided by Krippendorff [10]. One of the most commonly used measures for agreement between raters is Cohen's Kappa, which is used for two raters, and Fleiss' Kappa, which can be used for more than two raters [6]. In this study, Krippendorff's alpha is used instead as it is the most general measure of agreement, with appropriate reliability interpretations in content analysis [10]. According to Krippendorff, a score of 63% is the minimum score for slight agreement while a score above 80% indicates good agreement. A score of 80% means that the agreement is for 80% of the messages bigger than chance.

No email-related report could be found by the author and only one report—by Wolf et al. [22]—could be found where human performance is evaluated on a machine learning text dataset. The report by Wolf et al. [22] assesses the classification accuracy in different experimental situations on a standard document classification dataset. The experiment is performed on the 20newsgroup dataset. The goal is to evaluate how time pressure and the representation of the text as a BOW model influence the accuracy. They report that humans performed very well even with limited time and seeing only a BOW representation of the document.

2.2.2 Algorithm performance measurement. Apart from the evaluation of the human performance, the algorithm performance needs to be assessed. To compare the performance between different classification methods, standardized measures are used. An overview of the common measures for classification is given by Sokolova

et al. [20]. For the multi-class classification task in this study, for each class precision, recall and f1-score are relevant measures to understand the performance and the practical implications. For the total performance, the micro-averaged performance is considered over the macro-averaged performance. These standard measures are all described by Manning et al. [14], including the advantages and disadvantages of specific methods. If the emphasis is on the performance of large classes, micro-averaging is preferred. This is the case for the specific task in this study based on the company's organization, where, for less frequent contact reasons, lower processing capacities are available in the departments.

2.3 Automation of customer service communication

Automated email classification in the specific case of customer service emails has been studied by Van den Poel and Coussement [2]. They try to split a binary classification into complaint and non-complaint categories by using a set of handcrafted linguistic features. Another study [8] focusing on analysis of customer inquiries describes an approach to classify customer reports automatically with a lot of effort in manual feature creation. In general literature, for the specific domain of customer service, email classification is scarcely mentioned. In contrast, less application-area-specific literature for general email classification is common, even though the majority of research in the area of email classification covers spam or phishing email detection [16], which is a binary classification problem. With the popularity of deep learning approaches and social media, chatbots for customer service were introduced. Xu et al. [23] describe an approach for filtering out messages that are only emotional and do not include an actual problem. The documented approach is interesting because a content analysis section is included to compare between humans.

3 METHODOLOGY

3.1 Description of the data

3.1.1 General dataset information. The dataset used has been provided by Elsevier and contains Scopus customer service email messages. Scopus is a large abstract and citation database for academic publications. The dataset includes email correspondence of several months. A conversation can comprise multiple emails or only one initial message. The focus in this study is only on the first messages sent to open an incident in the customer service system. The dataset for each incident contains a contact reason, which is used for the company's internal problem-solving and problem-routing. There are multiple levels of contact reasons, but only the first one is considered in this study. Hence, the first contact reason is only referred to as contact reason. Emphasis has been put on protecting the customer by conducting the research GDPR compliant. Additionally, Elsevier's critical business information is protected by anonymizing contact reasons with the letters A-H and queues by the numbers 1-3.

In Table 1, an overview is provided about the size of the dataset. There are 23,168 incidents included in the dataset, but only 23,052 have been completed. Only the completed incidents are used as the assigned contact reason and the problem is only fully known after an incident has been completed. The 23,052 completed incidents represent the number of first messages, which can be used to assign a contact reason based on the first messages' text body. Each email falls under one of eight contact reasons. Each contact reason is processed by one of the three specialized departments (queues). An overview of the queues and their corresponding contact reasons is provided in Figure 1. Additionally, the figure shows the absolute number of messages per queue and per contact reason, as well as the percentage of total messages. The dataset is very imbalanced regarding the class distribution. There is one dominating class, with 48% of the total messages. The smallest class only contains 2% of the total number of messages.

Table 1: Summary of important figures about the email dataset used within this study.

Description	Unique items
Number of incidents	23168
Number of completed incidents	23052
Number of contact reasons	8
Number of service queues	3
Corpus size in words (Unigrams)	172314

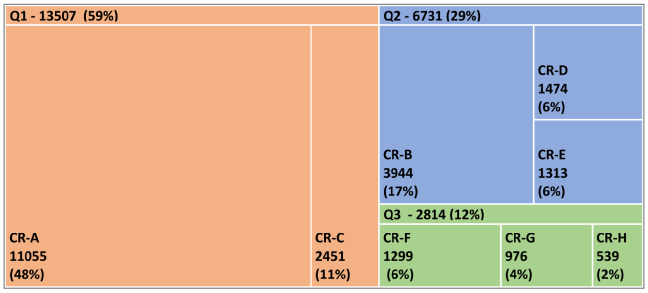

Figure 1: Overview of queues Q1-Q3 and the corresponding contact reasons CR-A to CR-H. The absolute numbers of first messages and the percentage of the total number of first messages are given.

3.1.2 Text message specific information. For each text message, there is a field containing the actual message and one contact reason label. From the contact reason label, it is possible to infer the queue directly as the corresponding queue for a contact reason is fixed. In 2, an example is given of the beginning of a text message and its contact reason. The email message content is the only information available, from which all features have to be derived for predicting the corresponding contact reason of the message. The text is the raw email, including HTML tags and headers or footers. A lot of variation regarding the structure is possible based on the email clients used by the customers and whether the emails were forwarded. The contact reason label in the data is the label which is assigned finally by the customer agent who resolved the problem. Between the initial message and the resolution, there could have been follow-up communication to elaborate the exact problem in more detail. Hence, this given label is more correct regarding the

customer's actual issue but does not always correspond to the text in the first message if it was not clearly described there.

Table 2: Important features listed in email dataset with an example value.

Field	Example Value
Thread Text	Dear Sir/ Madam, My institution (Universitat, Spain) is subscribed to ...
Contact Reason	Using the product

To gain an overview of the text messages regarding length and vocabulary, corpus size statistics and message length statistics are calculated. The emails are tokenized in words respectively unigrams after the HTML tags are removed to obtain the corpus size of the actual content. The corpus size when all first messages are used equals 172,314 unigrams. Plotting the word frequencies in relation to their frequency rank shows that the occurrences of the word is close to a power law distribution. The corresponding plot is illustrated in Figure 6 in the appendix A.

For the text length analysis, each contact reason is observed separately and for all contact reasons together. The average length, standard deviation, 25% quantile, 50% quantile, and 75% quantile are calculated and illustrated in Figure 2. All the messages together contain on average 197 unigrams. There is a visible difference in terms of the length distribution by class. The differences are visible mainly in the longest 25% to 50% of the messages. The contact reasons G, D, and F have significantly longer messages included. If the total of messages is observed, 50% of the messages are less than 112 unigrams long. Hence, the mails have a short text body in general if it is taken into account that the emails contain footers as well.

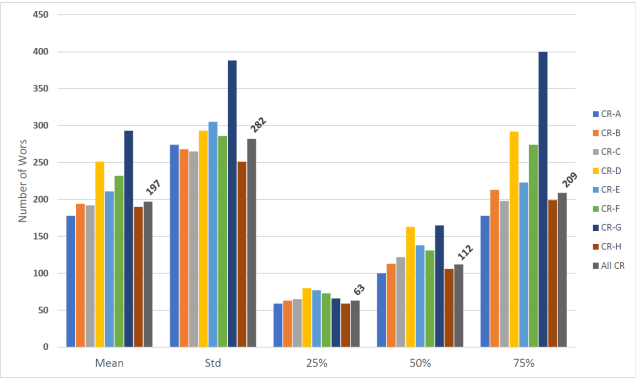

Figure 2: Illustration of text message lengths in words for each contact reason and the total of all messages.

3.1.3 Evaluation set. The validity of the training dataset described in Section 3.1 was assessed by handing out an evaluation set of 500 messages to two customer service agents from different queues, who assign contact reasons to emails on a daily basis. All the evaluation messages were already labelled in the system dataset with a contact reason. Random messages were selected based on the system dataset contact reasons. The contact reasons [A,B] are represented by 85 examples in the dataset and the contact reasons C-H by 55. The customer service agent had to assign a contact reason and a queue label for each message by only seeing the extracted text of the message. In a comment field, additional remarks were captured to provide information for the validity assessment. The number of 500 messages was determined as the maximum capacity that can be handed out to Elsevier's customer service agents for human annotation. As the original imbalance in the dataset is very high, a stratified sample of 500 messages would not lead to enough samples for assessing the less frequent class performance. Hence, the choice for a nearly balanced evaluation set was made, where only the contact reasons A and C are represented by 35% more samples to achieve balance between queues.

The service agents read the emails and provided comments if they were unable to assign a contact reason. Based on the feedback, 36 messages from the evaluation set, which were spam or did not include any information about the problem in text form, were removed. The remaining 464 messages were identified as messages with a text body, which is to some extent related to a customer service incident. Because the evaluation set was randomly picked for each class, a similar number (7%) of non-usable email messages is expected in the full dataset for each contact reason. The contact reasons F to H show a relatively high amount (10-15%) of messages, where no relation with the given contact reason could be found at all.

3.2 Methods

In this section, the methods used to gain insights into the research questions are described. The methods have two main areas of focus. The first area focuses on the methods used to assess human performance and make it comparable to the second part, which covers the automated classification approach. The human classification part describes what is considered as human performance in order to be compared with the automated performance. The automated classification part contains three major steps—extracting the text body, building a baseline model, and searching for improvements. At the end, the best performing classification model is picked and the agreement measure is calculated to compare with the human performance.

The general approach is illustrated in Figure 3. From the original dataset, 500 messages are extracted for human evaluation and classifier evaluation. The human evaluation is used to assess the difficulty of the task for humans based on the text body and to assess the validity of the dataset. The remaining data is used to train and evaluate classifiers. At the end, the results are compared between the best classifier and humans.

3.2.1 Text body extraction from email. The basis for evaluation and classification of an email in this study is the text body. The emails in the dataset are stored in the raw HTML structure created by the email client from which it was sent. Hence, a lot of noise and variation is included in the raw text of a message. The most important factors for variation are the email client used, the style

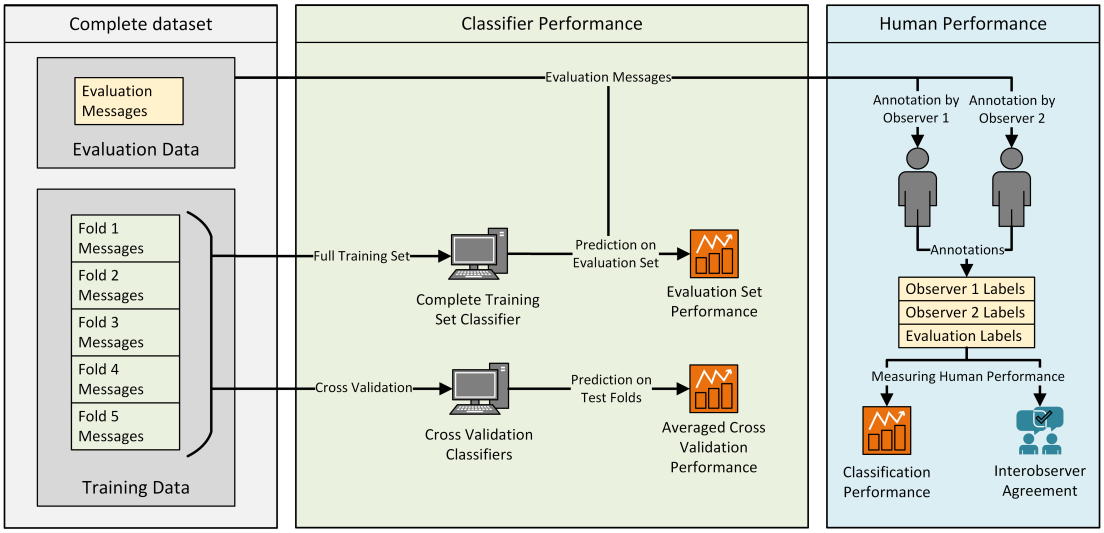

Figure 3: Diagram of the experiment setup. Extracting an evaluation set from the original dataset and handing it out to two annotators. The remaining data is used to train and evaluate the classifier by five-fold cross-validation. At the end, the performance results are compared.

of structuring an email body, and the footer with contact information. To remove critical information and reduce the noise, multiple cleaning steps are performed. The individual steps are explained in the appendix B.1.

The efficiency of the steps to extract and prepare the text body for human evaluation and classification are assessed by checking 100 messages manually. For each message, the original message is compared with the extracted text body. Then, the decision is made regarding whether the step was successful or not. The step is considered as successful if the text body with the problem description is still within the message after the extraction.

3.2.2 Service agent data evaluation and human performance. This section describes the evaluation procedure for the label validity and the human performance measurement.

Service agent data evaluation. The validity of the training dataset is assessed by measuring the agreement between the two observers (customer service agents) on the evaluation set. Krippendorff's α [10] score is used to measure the agreement between the observers. The Alpha score is the ratio between the observed disagreement and the expected disagreement by chance is subtracted from one. Perfect agreement is represented as $\alpha = 1$, no agreement $\alpha = 0$ and systematic disagreement with a negative α. A detailed explanation is provided by Krippendorff [11]. The nominal difference function is used and calculated with the NLTK [1] implementation. The agreement between the two observers and between each of the observers and the system dataset is calculated. The agreement score comparison between the observers itself and the observers with the system dataset is part of the assessment to decide if the system dataset labels are usable for training. If the agreement is good between the two observers ($\alpha > 80\%$) and is simultaneously low ($\alpha < 63\%$) between the observers and the system dataset, then doubts can be raised on the correctness of the system labels. Further investigation would be required as well in the case of extremely low agreement scores in general ($\alpha < 40\%$). The ranges for agreement are based on statistical analysis of Krippendorff [10]. It is important to be aware that for the system dataset labels, additional information—like the full conversation history and attachments—is available and can be viewed.

Human performance. Human performance is evaluated using two different approaches. The first approach is to use the α described in the previous paragraph. Including the system dataset, which is also created by human service agents but is considered to be more correct, there are three human-labelled contact reasons available. Hence, the total agreement between all three datasets and between the two observers is used as a human performance score. The second approach is to calculate precision, recall, and f1-scores for each of the two observers on the evaluation set by using the system dataset as a gold standard. The total performance is measured by the micro-averaged f1-score. In the case of the evaluation set, the micro-averaged score is nearly a macro-average due to the balanced representation of the classes in the evaluation set. Confusion matrices are used to analyze the errors on contact reason level classification and queue level.

For the specific task of labeling customer service messages, a high precision score implies that a person is assigning a certain contact reason or queue only when the person is sure about the label. A high recall metric means that the messages that should be assigned to a specific contact reason or queue are nearly always identified. For the specific case of the Elsevier customer service structure, it is important achieve a high recall on the biggest classes because otherwise too many messages are routed to smaller departments, which do not have the capacity to handle many messages, resulting in a big backlog.

[1] https://www.nltk.org/_modules/nltk/metrics/agreement

3.2.3 Text classification baseline setup. To establish a strong baseline for the automated classification a linear SVM is used and the input features are represented in a BOW model by using unigrams with term frequency-inverse document frequency (tf-idf) weighting [14]. The training dataset is split into five folds and cross-validation is used to evaluate the performance. The 500 messages used in the evaluation set have been excluded previously. A brief overview is provided in Figure 4. Each individual step used for the baseline classifier is described in more detail in the following paragraphs.

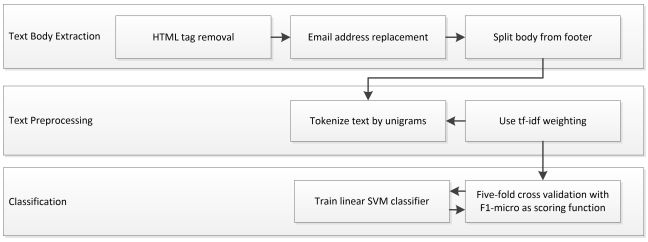

Figure 4: Illustration of the steps included in the baseline classifier.

Preprocessing. The start of preprocessing is considered within this document after the text body extraction has been performed, as described in Section 3.2.1. The difference is that the text body extraction is seen as a necessary step to create a usable dataset for human and automated classification. The preprocessing is only used for preparing the data for automated classification. Hence, the extracted text body is used in the first step of preprocessing. To create features that can be used by an SVM, a vector representation of the text body has to be created. There are various methods that can transform text into a vector representation. The most frequently used one is the representation of the text as a BOW, where each element in a vector represents a word or an n-gram. Hence, first the n-grams have to be created by tokenizing the text into n-grams. The n-grams are created by using a word tokenizer, which has the objective of separating words or word combinations within a document. In the baseline, unigrams are used and the tf-idf weighting is used instead of raw term frequencies. The tf-idf weighting has the main advantage that the impact of a word is dependent on how frequently it occurs in other documents. Hence, it is assumed that words occurring in all documents or many documents are less discriminating for the classification of a document to a category than words occurring in only in a smaller number of documents. For example, the word Scopus is expected to be present in many of the emails because all email messages are about Scopus, but it does not provide any information about which contact reason an email is written. These two steps are implemented by using the standard configuration of the scikit-learn tfidfvectorizer [2] by using the default configuration.

Classifier. For the multi-class classification problem of assigning contact reasons to messages, a linear SVM is used. The linear support vector classifier has multiple hyperparameters. The first parameter, referred to as the C value, influences the number of datapoints within the support vectors when the decision boundary is fit. The standard setting of parameter C is the value 1.0.

An SVM is by definition a binary classifier. Therefore, for training a multi-class model, multiple classifiers have to be trained. There are two commonly used training strategies. The first one is the one-vs.-all strategy, where one specific class, e.g. the contact reason A, is the first class and all other contact reasons B-H are the second class. Using this strategy requires the training of one classifier per class. The other common method is to use a one-vs.-one strategy, where a classifier is trained for each class pair. This strategy requires the training of more classifiers but can lead to better or faster results under some conditions. For example, a one-vs.-one strategy can be beneficial if there are many small classes and only one much bigger class. For the baseline, the one-vs.-all approach is used with the linearSVC implementation of scikit-learn.

The model is evaluated and trained using a five-fold cross-validation on the full training set without any adjustment on the sampling procedure. The distribution of the class frequencies is therefore imbalanced, as in the complete dataset.

3.2.4 Algorithm performance measurement. The classifier performance is evaluated in two different situations. The first one is the performance on the test set during cross-validation, which is the reference for comparing the automated models. The second one is the performance of the evaluation set, which is used to obtain a directly comparable performance on a small set between the automated predictions and the human annotations. The measures calculated are the same as described in 3.2.2 for the human performance. The only difference is regarding the α score, which is only calculated between the original system label and the predicted label. The F1 micro-averaged scores are used as scoring function because good performance on the biggest class is important. In comparison to the evaluation set on the training set, there is a big difference between the micro- and macro-averaging scores. The same rules regarding the importance of the recall on the most frequent classes apply as for the human performance.

3.2.5 Algorithm improvements. In this section, the attempts to improve the performance of the classifier compared to the baseline are described. For text classification, many steps can be adjusted within the pipeline, which might lead to better classification results. A better classification result is defined as the result with the highest micro-averaged F1 score; if the F1 scores on the contact reason are equal, the best F1 micro-score on the queue level is used for selection. The search for improvements is structured into multiple parts, each related to a specific step within the pipeline. In this part, the improvements are briefly described a more detailed description for each paragraph can be found in the appendix B.2.

Sampling adjustments. The number of messages for each contact reason class in the training data is adjusted to balance the examples fed to the classifier. Oversampling and under-sampling are two commonly used techniques. Under-sampling reduces the examples used for the most frequent classes and oversampling increases the number of examples of the less frequent classes. Under-sampling leads to a situation where not all available information is used, which could be used for training a classifier. Oversampling leads

[2] http://scikit-learn.org

to an over-representation of individual text messages because the sample message is fed to a classifier multiple times. A better balance can lead to less bias toward the most frequent classes.

- Under-sampling. Decreasing the number of samples in the majority class.
- Oversampling. Increase the number of samples in the smaller classes by drawing with replacement.
- Combine over- and under-sampling.

Preprocessing adjustments. The preprocessing of the data is adjusted to change the text representation fed to the SVM classifier. In the baseline, the words of the text were used unchanged and only unigrams were used. Techniques like stemming or lemmatizing reduce the words to their stem or base form. Stemmed words are no longer actual words; the stems are rule base created without a dictionary. Lemmatizing retains the actual words and reduces them to their base form by using a dictionary. Part-of-speech (POS) tags are used in combination with lemmatizing to reduce the word from its inflectional form to a base form while retaining the original part of speech.

- Stemming
- Lemmatizing
- Lemmatizing with Part-of-speech tags
- Use unigrams, bigrams, trigrams

Hyperparameter tuning of support vector classifier. For an SVM, a few hyperparameters can be adjusted. In the linear case, the main parameter is the C value and in the case of using a radial basis (RBF) kernel, an additional parameter, gamma, is adjustable. The training method can vary between the one-vs.-all and one-vs.-one training strategies.

- Adjust kernel from linear to radial basis function (RBF)
- one-vs.-all and one-vs.-one
- Adjust C

Ensemble. In the ensemble section, classifiers other than the SVM are used for the first time. A voting method is used based on the majority vote with different classifiers and a bagging method with only SVM. The SVM are only trained on a smaller subset of the data in the case of bagging. Furthermore, an approach is attempted where four individual classifiers are trained. The first classifier predicts the queue and then—for each individual queue—a classifier predicts the contact reasons possible within the queue. Combining multiple classifiers, according to the literature, can lead to better performance than a single classifier itself [1].

- Voting logistic regression, naive Bayes and random forest
- Bagging SVM
- 4 individual SVM Classifiers. First queue prediction and then contact reason.

LSTM. As a last improvement step, two RNN architectures are evaluated. Specifically, an RNN is used with LSTM cells in a unidirectional architecture and a bidirectional architecture. According to the literature, they can perform well on short text classification and have gained popularity over the last few years [5].

- One directional LSTM
- Bi directional LSTM

3.2.6 Final human and automated comparison. The final comparison of the performance between the human performance and the automate performance is made based on the comparison of the performance of the two annotators with the baseline model and the best performing improved model. For each automated model, there are two performances—one is obtained during cross-validation and the other by predicting the labels on the evaluation set. The same statistics and measures introduced for the human performance and the baseline performance in the previous sections are used. The rules for comparison are that the performance on the bigger classes is more important than that on the less frequent classes. The performance is compared for each contact reason and each queue.

4 EVALUATION

4.1 Text body extraction

In an experiment with 100 messages, for each message, it was assessed whether the text body extraction steps worked. The results of the evaluation are shown in Table 3. The first step of the text body extraction includes the removal of header-specific HTML tags and the extraction of the email text without the HTML tags. The package "Beautiful Soup" [3] achieves very good results on the email messages; even many emails have a broken HTML structure. The results listed in Table 3 are assessed regarding the goal of keeping the important email body, which includes the user-written problem description to customer service. All the HTML text was correctly extracted; in two cases, the removal of Gmail signature and quote tags caused a removal of the problem description. All tags where removed if they had to be removed and 98% were correctly removed. The same approach has been used to evaluate if the text body is separated correctly from the footer or chained emails. A few naive rules have been used to remove footers from the text body. The footer signatures have been removed in 74%, if there were any, and 97% of the removed content was correctly removed. Hence, the important text body content with the description of the problem remained in nearly all of the cases.

Table 3: Text body extraction success overview.

	Text Extraction	Split Text Body
Removed correctly	87	58
Remained correctly	11	20
Removed too much	2	2
Removed not enough	0	20
Removal precision	**98%**	**97%**
Removal recall	**100%**	**74%**

4.2 Service agent data evaluation and human performance

4.2.1 Service agent data evaluation. The agreement on the label between the two observers itself and between the observers and the system dataset is listed in Table 4. All the α scores are between 46% and 50% on the contact reason level. On the queue level, the

[3] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

agreement is better with a range from 60% to 65%. The agreement on the contact reason is low for the dataset; the agreement is bigger than chance only for 50% of the data. On the queue level, the agreements are better and slight agreement is measured according to the definition of Krippendorff [10].

Table 4: Agreement on contact reasons and queues in human annotated datasets.

Set 1	Set 2	Contact reason α	Queue α
System Dataset	Observer 1	0.46	0.61
System Dataset	Observer 2	0.50	0.60
Observer 1	Observer 2	0.49	0.65
All three		0.49	0.62

These results are definitely not a high agreement and show that the task of identifying the contact reason from text is not trivial for humans. More important than the degree of agreement is the observation that the agreement between the two human-annotated evaluation sets is not significantly better than between the human-annotated datasets and the system-exported dataset. This observation does not conflict with the assumption that the dataset in the system is more correct than the single human assessed. If there would have been perfect agreement between the two raters and bad agreement toward the system datasets, further investigations would have been needed to assess why the labels are that different to the system values.

4.2.2 Human performance. The classification results of the human annotators are illustrated in Figure 5 with a line for each performance measure and each target variable. The performance measures shown are precision, recall, and f1-score on a queue routing basis and a per contact reason basis. The total micro-averaged performance scores for the queue and contact reason level are listed in Table 5. The total agreement between the observers is shown in Table 4. The total micro-averaged f1-score, which is due to the hardly imbalanced evaluation set nearly a macro-average is 74% for Observer 1 and 73% for Observer 2 on queue classification. For the contact reason classification, 51% and 57% are measured.

Comparing Observer 1 and Observer 2 on a contact level shows that they are close together on precision and recall for contact reason A, which is the biggest class. On all the other contact reasons, except the least frequent class H, the performance differs significantly on either recall or precision. Class H has in general, extremely low performance, which could be due to various reasons. One observer is not better than the other on all the classes. Nevertheless, Observer 2 has a more constant performance over all classes. Annotator 1 was significantly worse in terms of identifying the affiliation profile correction and citation correction class. The content class could not be identified at all by the first annotator. The extremely low performance on contact reason H could have various reasons, such as ambiguity with another class from the taxonomy, no clear definition, or insufficient agent training.

If the focus is solely on the queue to which an email message would have been routed based on the annotator's label, then all the metrics are higher than 60%. The messages from the Queue 1 are identified by both annotators in a similar range, with slightly better performance by Observer 1. Queue 2 shows a big difference in recall between the observers. Observer 1 has identified over 80% of the messages that belong to Queue 2. For Queue 3, with the least volume, Observer 1 is precise at assigning the queue but has a significantly lower recall than Observer 2. Having a high precision is important in practice for the small queue to not overload the low capacity queue with many wrong messages. Hence, the performance of Observer 1 is preferred for Queue 3. The observations support the presumption that Observer 1 is better at routing Queue 1 and Observer 2 is better at routing Queue 3. The assumption was made based on the department in which Observer 1 and Observer 2 work. Nevertheless, the differences observed are much smaller than expected.

4.2.3 Human errors. In Figure 7 and Figure 9 of the appendix C, the confusion matrix of the errors is presented. The human errors can be identified in the matrices, which show that the errors were made in similar areas by the two annotators. The contact reasons G and F were confused most often. The contact reason H was often misinterpreted as contact reasons A and C. In general, for both annotators, a slight bias toward the contact reasons A and C is observed. Observer 1 shows a stronger bias toward A and Observer 2 toward C.

4.3 Text classification baseline results

The results of the baseline classifier are illustrated in Figure 5 in the same way as the human performance. In contrast to the human performance measures, there are two lines for the same baseline classifier. The lines differ by the data on the basis of which the predictions have been made. The first line uses the exact same messages of the evaluation set as for the human performance measurement. The second line is calculated based on the averaged cross-validation test scores of the full training dataset. The results on the evaluation set show for the baseline a low precision on contact reason A—only a little over 40% . For the cross-validation training set, the precision is close to 75% for the same contact reason. The recall is close to 90% in both variants. Most of the classes are in a bandwidth regarding precision and recall from 55% to 65%, except contact reason D, which has very good precision and recall scores, and contact reason H, which has extremely poor precision and recall scores. The total f1-score of the baseline classifier for the evaluation set is 57% on the evaluation set and 70% on the training set. In the case of the evaluation set, nearly a macro-averaged performance is measured due to the balance in the contact reasons. The small classes are over-represented in the evaluation set compared to the daily business pattern within the training set. Measuring and comparing the performance on both datasets offer better insights into how well the classifier is able to distinguish between the classes. The second line is the performance, which can be expected in practice if the messages are received in similar proportions with similar content in the future.

The results on the queue level measured by cross-validation are very good. A total f1-score of 81% is measured at training and 74% on the evaluation set. On the queue level, the difference in precision for Queue 1 between the evaluation set and the training data is high. The f1-score performance is stable over all the queues. The results on the queue level show a favorable pattern for the precision and

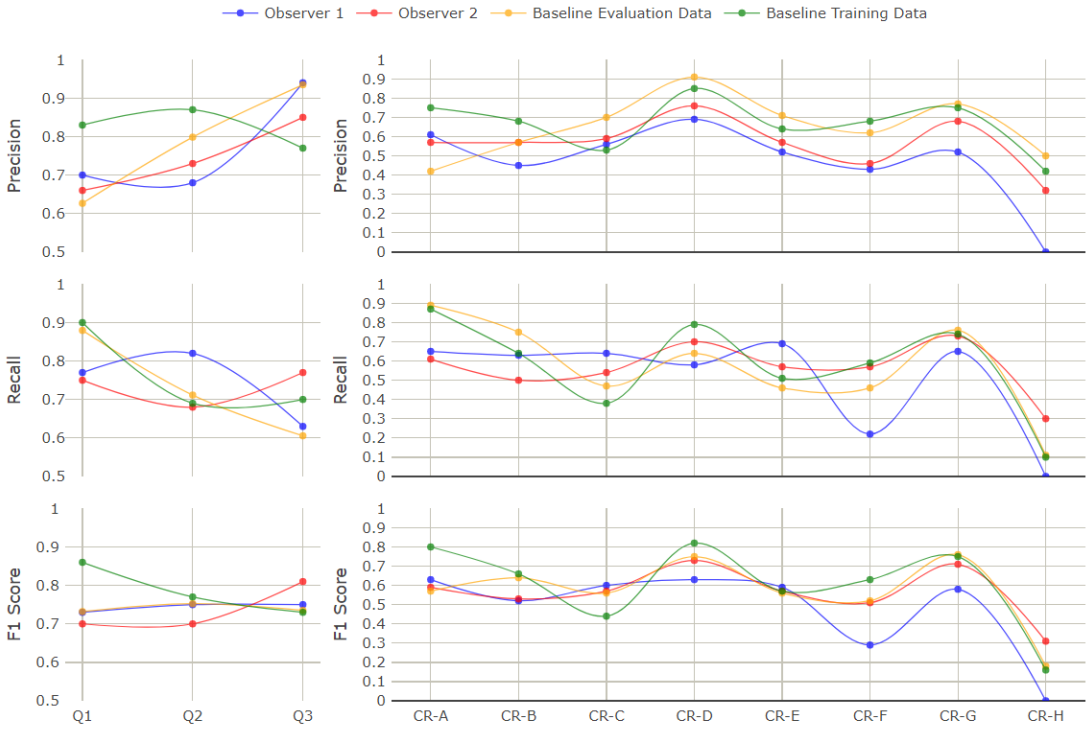

Figure 5: Performance comparison on contact reasons level and queue level. Observer 1, Observer 2, and classifier evaluation data are all evaluated on the same evaluation set. Classifier training data represents the average scores of the cross-validation.

Table 5: Total performance for the observers on the evaluation set and the classifier on the training set.

Total scores	Precision Q	Precision CR	Recall Q	Recall CR	F1 Queue	F1 CR	Alpha Q	Alpha CR
Observer 1	0.74	0.53	0.74	0.53	0.74	0.53	0.61	0.46
Observer 2	0.73	0.57	0.73	0.57	0.73	0.57	0.6	0.5
Classifier Evaluation	0.78	0.64	0.74	0.59	0.74	0.57	0.6	0.51
Classifier Training	0.82	0.70	0.82	0.72	0.81	0.70	0.66	0.58

the recall for practical business application. The precision increases for Queues 2 and 3 with less volume and the recall is highest for Queue 1 with the highest message volume.

In Figure 8 of the appendix C, the errors of the classifier can be analyzed in the confusion matrix. The performance on the evaluation set and on the cross-validation test set is compared. The confusion matrix clearly shows a strong bias toward contact reason A and a small bias toward the B class, which are the largest classes. Additionally, the contact reasons G and F are likely to be confused. If the results on the queue level are observed, there is a visible bias toward Queue 1 class. Other confusions are low and therefore most misclassified messages end up at Queue 1. Furthermore, the distribution of the cross-validation performance scores is illustrated in Figure 11 of the appendix C.

4.4 Text classification improvements results

During the search for improvements, no significant performance gains were found regarding the total performance. The baseline model provides the same performance as the improvements with a simpler classification approach. Hence, the baseline model remains the model for comparison with human performance in the next section. Results for a selection of the different models are documented in the appendix B.2. This section summarizes the most important insights.

A major observation from the search for improvements is that there are contact reasons that can be hardly identified in any model configuration. The performance observed for many of the improvements is very close to the baseline model even, though a completely different classification approach is used. The classification results are in general stable and not very dependent on the exact input structure or a single parameter of the classifier. The contact reason H cannot be reliably identified in any of the models. The contact

reasons F and G are often confused and none of the improvements can reduce the confusion between them. The same problem applies to the contact reasons A and C. The total performance could not be improved, but with under-sampling, the emphasis on precision or recall for the most frequent class A could have been influenced strongly.

5 CONCLUSION & DISCUSSION

5.1 Conclusion

Measuring the human performance of the service agents (observers) in labeling the messages with a contact reason, showed the task is for humans challenging. The agreement between the two observers was low at assigning contact reasons and slight on queue routing. Training a baseline classifier lead to a classifier which performs in a similar range as the humans. Searching for improvements by adjusting the classifier and its input, did not lead to a significant performance gain. Hence, the baseline model is used for comparison with human performance to answer the main research question.

The baseline model performed better or equally compared to the human service agents. The baseline model reached a total F1 score of 57% on the evaluation set. This score is equal to or better than those of the observers, who scored 53% and 57% for the contact reason classification. For the queue routing, the baseline model performed equally well as the observers, with a 74% F1 score. If the performance of the classifier measured during cross-validation on the training set is considered, the classifier performs significantly better than the human service agents. F1 scores of 70% for the contact reason prediction and 81% for the queue routing are measured. All the total scores listed in Table 5 are in favor of the classifier.

In an individual comparison, the classifier could keep up with the human service agents or outperform them. The results are not as clear as for the total performance but the individual performance comparison shown in Figure 4.3 leads to the conclusion that the classifier can perform in a similar range or better on seven out of eight contact reasons and all the queues. Hence, the human performance can be reached using automated text classification methods at the task of labelling customer service emails with contact reasons by only using the text body.

5.2 Discussion

The most important insight regarding customer service automation in general and especially for Elsevier is, even with a standard text classification approach a similar or better performance than for the service agents can be possible. The expectations in the improvement methods were clearly not met. None of the expected experiments lead to significant improvements. During the research process it became obvious that is hardly possible to generalize form one dataset to another, even though they have similar characteristics. To achieve the automation of the task in practice in future, a bigger evaluation set should be annotated by at least one more human annotator, where the original class distribution is retained. This could lead to a better assessment of the current human performance, to which the classifier can be compared. The micro-averaged F1 scores can only be properly compared in a dataset with the original label imbalance. Furthermore, not labeling spam emails with a contact reason or filtering them out from the training set could help to improve the training set quality and therefore lead to a better classifier and more accurate test results. Additionally, the current business process needs to be assessed thoroughly. The percentage of messages which are updated during the incident solving is not known. Hence, it's not clear how many messages are actually labelled the first time wrong. Furthermore, the impact of a wrong labelled message regarding processing time, customer satisfaction and operating cost needs to be assessed to evaluate advantages and disadvantages of an automated system.

A practical application of the classifier could be a system where the human agent and the classifier co-exist. The messages where the contact reason can be classified with high confidence should be automatically sent to the relevant department. Remaining low confidence messages are still processed by customer service agents manually. Such a setup can lead to a significant reduction of the workload for the customer service department and speeds up the processing of many standard inquires. Additionally, it is possible to use the classifier to provide the customer service agent suggestions for a label. The approach of automatic classification could even be extended to Elsevier's online webforms for Scopus support, where currently the user has to choose a contact reason by himself. An automatic system could provide the customer a suggestion based on his text and eliminate the additional step for the customer. Further, research could be conducted in the direction if it is possible to answer some of the questions automatically.

ACKNOWLEDGMENTS

I would like to thank Elsevier for providing the opportunity to work on this interesting project. In particular, I am grateful to Zubair Afzal for supervising me internally in the company and providing valuable inputs and technical knowledge. Furthermore, I would like to thank Maarten Marx for the helpful feedback and suggestions during the project.

REFERENCES

[1] Aggarwal, C. C., and Zhai, C. X. *Mining Text Data*, first ed. Springer, New York, NY, 2012.
[2] Coussement, K., and den Poel, D. V. Improving customer complaint management by automatic email classification using linguistic style features as predictors. *Decision Support Systems 44*, 4 (2008), 870–882.
[3] Ede, G. How automation should be mixed with human contact for ideal customer service. *Admap Magazine*, September (2017).
[4] Estabrooks, A., Jo, T., and Japkowicz, N. A Multiple Resampling Method for Learning from Imbalanced Data Sets. *Computational Intelligence 20*, 1 (2004), 18–36.
[5] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.
[6] Hallgren, K. A. Computing Inter-Rater Reliability for Observational Data: An Overview and Tutorial. *Tutorials in quantitative methods for psychology 8*, 1 (2012), 23–34.
[7] Istanbulluoglu, D. Complaint handling on social media: The impact of multiple response times on consumer satisfaction. *Computers in Human Behavior 74* (2017), 72–82.
[8] Jetley, R. P., Gugaliya, J. K., and Javed, S. Using Text Mining for Automated Customer Inquiry Classification. In *The Fifth International Conference on Business Intelligence and Technology* (Nice, France, 2015), pp. 46–51.
[9] Joachims, T. Text categorization with Support Vector Machines: Learning with many relevant features. In *Proceedings of the 1998 European Conference on Machine Learning* (Berlin, Heidelberg, 1998), Springer Berlin Heidelberg, pp. 137–142.
[10] Krippendorff, K. *Content Analysis: An Introduction to Its Methodology*, second ed. Sage Publications, Thousand Oaks, CA USA, 2004.
[11] Krippendorff, K. Computing Krippendorff's alpha-reliability. 2011.
[12] Legett, K. Customer Service Trends : Operations Become Smarter And More Strategic. Tech. rep., Forrester, 2017.

[13] Ma, X., and Hovy, E. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (Berlin, Germany, 2016), vol. 1, pp. 1064–1074.
[14] Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA, 2008.
[15] Mirończuk, M. M., and Protasiewicz, J. A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications 106* (2018), 36–54.
[16] Mujtaba, G., Shuib, L., Raj, R. G., Majeed, N., and Al-Garadi, M. A. Email Classification Research Trends: Review and Open Issues. *IEEE Access 5* (2017), 9044–9064.
[17] Pennington, J., Socher, R., and Manning, C. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), pp. 1532–1543.
[18] Rifkin, R., and Klautau, A. In defense of one-vs-all classification. *Journal of machine learning research 5*, Jan (2004), 101–141.
[19] Sebastiani, F. Machine Learning in Automated Text Categorization. *ACM Computing Surveys 34*, 1 (2002), 1–47.
[20] Sokolova, M., and Lapalme, G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management 45*, 4 (2009), 427–437.
[21] Windley, P. J. Delivering high availability services using a multi-tiered support model. *Windley's Technometria 16* (2002), 1–9.
[22] Wolf, F., Poggio, T., and Sinha, P. Human document classification using bags of words. Tech. rep., Massachusetts Institute of Technology, Cambridge, MA, USA, 2006.
[23] Xu, A., Liu, Z., Guo, Y., Sinha, V., and Akkiraju, R. A New Chatbot for Customer Service on Social Media. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2017), ACM, pp. 3506–3510.
[24] Yin, W., Kann, K., Yu, M., and Schütze, H. Comparative Study of CNN and RNN for Natural Language Processing. *arXiv preprint arXiv:1702.01923 abs/1702.0* (2017).
[25] Young, T., Hazarika, D., Poria, S., and Cambria, E. Recent Trends in Deep Learning Based Natural Language Processing. *arXiv preprint arXiv:1708.02709 abs/1708.0* (2017).
[26] Zhang, W., Yoshida, T., and Tang, X. A comparative study of TF*IDF, LSI and multi-words for text classification. *Expert Systems with Applications 38*, 3 (2011), 2758–2765.

A DATA

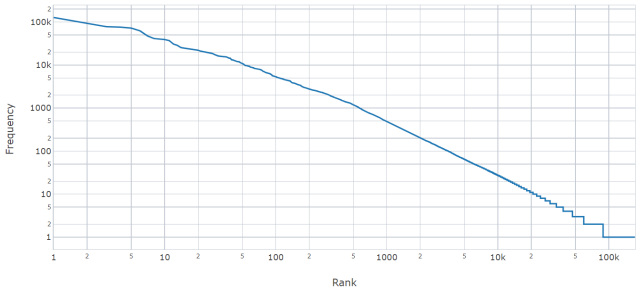

Figure 6: Word frequencies versus rank plotted on logarithmic scales for the whole corpus.

B METHODS

B.1 Text body extraction from email

To remove critical information and reduce the noise, the steps listed below have been executed on all of the messages.

(1) Use the Beautiful Soup package to parse the HTML content of the emails and remove the "gmail_signature" and "gmail_quote" tags. Furthermore, extract the text without the remaining tags.
(2) Replace the email address with a placeholder.
(3) Split the email by common character sequences for indicating a forwarded email. Additionally, split the email by common text endings rule based to keep only the first split without the footer.

The HMTL tags in the email are removed using the Beautiful Soup package to extract the actual message text. Two specific metadata headers used by the Gmail client are removed in this step as well. The HTML removal step is conducted to create a good readable text format for the human evaluation and to reduce the amount of information without reasonable predictable value. It is assumed that the use of HTML tags does not deliver justifiable performance improvements for assessing if a message belongs to a specific contact reason.

With the current organizational structure, each queue has its own email address for support. Those email addresses are often included in the customer emails and would already be a very distinctive feature for deciding which contact reasons apply, as every contact reason is linked to a queue. Keeping the email addresses in the text body would therefore lead to better performance on this dataset but is contradictory to a pooled system, where all the emails are processed through one centralized pipeline and then distributed to the queue. Additionally, changes in email addresses can occur. Therefore, the system should be independent of individual email addresses. To achieve this, the email addresses are replaced with a placeholder.

Another step is to separate the actual text body from the rest of the email. The separation is done because the classifier and the human annotator should not be biased by the name or institution of a request. The goal is to identify and assess the problem only on the basis of the problem description. Good generalization is expected only by the separation of the text body in an environment where customers and institutions can change. The separation of the text body can be a challenging task. In this study, the decision has been taken to use a simple rule-based separation. A small number of random emails have been checked manually for elements that separate the footer from the text body. There is a lot of diversity regarding the separation but some easy patterns have been identified. Greetings are used as separator or e.g. many minus characters which are used by people as separators between text and footer. The method is very naive and, therefore, its efficiency needs to be assessed. It is considered to be less severe to not cut off enough information rather than cutting off too much.

B.2 Improvements

B.2.1 Sampling. To evaluate the impact of under-sampling on the classification performance, the number of examples in the class A is gradually reduced. The reduction is performed with steps of 20%, 30%, and 50%. In the case of 50% reduction, the two biggest classes, A and B, are nearly equally represented. Changing the number of examples for a specific contact reason could tackle the classification bias in imbalanced datasets. In the case of the baseline model, it is obvious from the confusion matrix in Figure 8 that the classifications are biased toward the most frequent contact reason A. Using less of the data for the majority class to train the classifier can lead to better balance between the classes.

For oversampling, the messages are drawn with replacement for all of the classes except the biggest class A. The amount of oversampling is increased by 50%, 100%, and 200%. In the case of 300% increase, every message is fed to the classifier of the lower classes on average three times and the number of total messages in the biggest two classes is nearly equal. The total number of different messages cannot be changed in the training dataset; therefore, it is only possible to use the same pool of messages and feed them multiple times to the classifier.

In the last sampling step, a combination of the two is used. Hence, the 20% under-sampling is combined with the 50% oversampling and the 30% under-sampling with the 100% oversampling.

B.2.2 Preprocessing. The preprocessing in the baseline model uses only unigrams tokenized by word. As a part of searching preprocessing improvements, a stemmer from the NLTK package [4], called snowball stemmer, is used. The wordnet lemmatizer is used with and without making use of part-of-speech tags. The goal of all these methods is to reduce the inflectual forms to a common base form. The stemmer achieves it in a rule-based manner and the output is often no longer an actual word. The lemmatizer is dictionary-based and retains the full words. By default, the NLTK word net lemmatizer lemmatizes all the words to a noun. Therefore, the part-of-speech tags are used to keep the word in its original part of speech.

An additional part where the preprocessing can be adjusted comprises the parameters of the vectorizer and the term weighting. In this step, the parameters of the scikit learn TfidfVectorizer are adjusted and the best selection is made by performing a grid search. For the following parameters, the best combination is searched.

- Stop words removal English or no removal
- Use of unigrams, bigrams, trigrams
- Using the sublinear tf-idf option where the importance of the occurrences of a n-gram is not treated linearly.
- The minimum of occurrences of a n-gram (1,3, 5) to incorporate it in the vocabulary.
- Replacing numbers with a placeholder

B.2.3 Single SVM adjustments. A linear support vector machine trained with the one-versus-all approach has been used as a baseline classifier. As possible improvement, the training approach is changed to a one-versus-one approach.

Apart from the training strategy, the linear support vector machine has a parameter C, which can be adjusted. Increasing the value of the parameter C penalizes those datapoints more which are lying within the support vectors. Adjusting this parameter could lead to a better fit of the model. Hence, the values [1,3,5,10,30,100] are used to evaluate if there is a better fit possible by using a different C value.

Instead of using a linear kernel, it is possible to use a nonlinear kernel which could separate the data better if it is not linearly separable. A common nonlinear kernel is the radial basis function (RBF) kernel [14].

[4]https://www.nltk.org/

B.2.4 Ensemble methods. In all the previous classifications, a classifier has directly predicted the contact reasons. As an alternative, in this section, the queue is predicted first and the contact reason is predicted in a second step based on the result of the queue classifier. The idea behind this approach is that the messages are distributed approximately 60%, 30%, and 10% over the queues. For each queue, there are more total messages available to train a classifier than for a contact reason. Pooling those messages could lead to a better performance regarding the routing of the queue, which is an important factor in the business process to avoid backlogs and waiting times.

In this study 10,20, and 30 SVM classifiers are trained with a bagging approach on a subset of max 50% and 70% of the data. Furthermore, the combination of different classifiers to a voting ensemble is tested. Random forest, logistic regression, and support vector machine classifiers are used and the majority vote determines the label selected.

B.2.5 Recurrent neural networks. In the previous sections, mainly an SVM is used for the classification. This section focuses on a different and more recent approach of text classification. An unidirectional and a bidirectional LSTM is trained with global vectors (GloVe) as an embedding layer. A subset of the possible parameters is adjusted to search for improvements. The bidirectional configuration is listed in Table 6

Table 6: Bidirectional LSTM configuration used for the contact reason classification with Keras and Tensorflow.

Layer (type)	Output Shape	Parameters
Input	400	0
Embedding	(400, 200)	4000000
Bidirectional	(400, 128)	135680
Global max pooling	128	0
Dense	64	8256
Dropout	64	0
Dense	8	520

A recurrent neural network can process sequences of text and exploit information about the positions of words in a sentence. In a unidirectional approach, only the context before a word is taken into account when learning to classify. With a bidirectional recurrent neural network, the context after a word is taken into account as well. The LSTM cells are used to learn the short- and long-term dependencies of a word regarding its context. The GloVe representation is a dense vector representation, which can be self-trained or used pretrained from a large corpus. The vectors can represent semantics of similar words in a lower dimensional space.

C RESULTS

C.1 Improvements

C.1.1 Sampling. The results of gradually decreasing the amount of messages for contact reason A showed, that small changes in under-sampling are hardly noticeable. Only significant changes where the number of messages for contact reason A gets close towards the number of messages for contact reason B make a big

difference. For the oversampling very little impact in general is observed. The most extreme changes of the experiment are illustrated in Figure 12. Relevant to the business process is the observation that with under-sampling the trade off between high recall on the two biggest queues is adjustable. This is important if the number of missclassified messages should be adjusted between the two big queues.

C.1.2 Preprocessing. The results of gradually decreasing the number of messages for contact reason A show that small changes in under-sampling are hardly noticeable. Only significant changes, where the number of messages for contact reason A is close to the number of messages for contact reason B, make a big difference. For the oversampling, very little impact is observed in general. The most extreme changes of the experiment are illustrated in Figure 12. Relevant to the business process is the observation that with under-sampling, the trade-off between high recall on the two biggest queues is adjustable. This is important if the number of misclassified messages has to be adjusted between the two big queues.

C.1.3 Single SVM adjustments and ensemble. The tuning of the hyperparameter C resulted in keeping it equal to 1.0, like in the baseline. The results of changing the kernel to RBF show a bias toward the majority class. Even when the contact reason A class was under-sampled, the classifier always predicted the same contact reason A for all the data. Hence, the classifiers predictions were not usable. The adjustment of the training method to one-vs.-one increased the precision but the total result was slightly worse than the baseline.

The ensemble approach was not successful in general for the bagging and voting classifier. No noticeable performance gain was observed. The classification for the two levels, where first the queue and afterwards the contact reason are predicted, is nearly identical with the single baseline classifier.

The results of the training strategy and the two-level prediction are illustrated in Figure 14. Other steps are not illustrated as they overlapped too closely with the baseline performance or, in case of the RBF kernel, the prediction performance was zero for all of the classes except contact reason A.

C.1.4 Recurrent neural networks. The best results for the unidirectional and bidirectional LSTM are illustrated in Figure 15. The unidirectional approach showed a slightly lower performance than the bidirectional variant. The main difference between the two approaches in terms of performance is observed in precision. The F1 score for the most frequent contact reasons A-C is better than the baseline with the bidirectional LSTM.

The adjustment of the training and network parameters shows that increasing the complexity of the network quickly leads to overfitting. The adjustment of the GloVe size and text length parameter showed little influence.

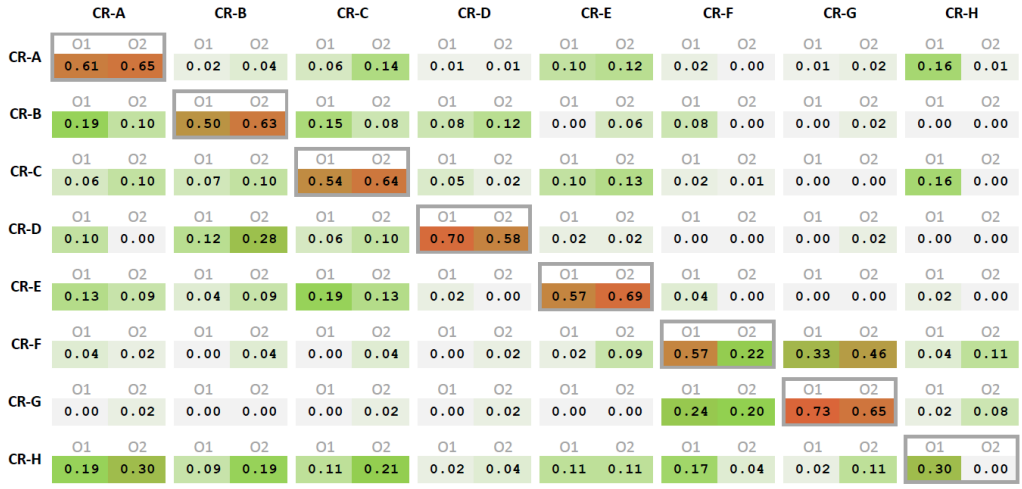

Figure 7: Human confusion matrix with fractions of total messages per contact reason. Comparison between the results of observer 1 (O1) and observer 2 (O2) on the evaluation set.

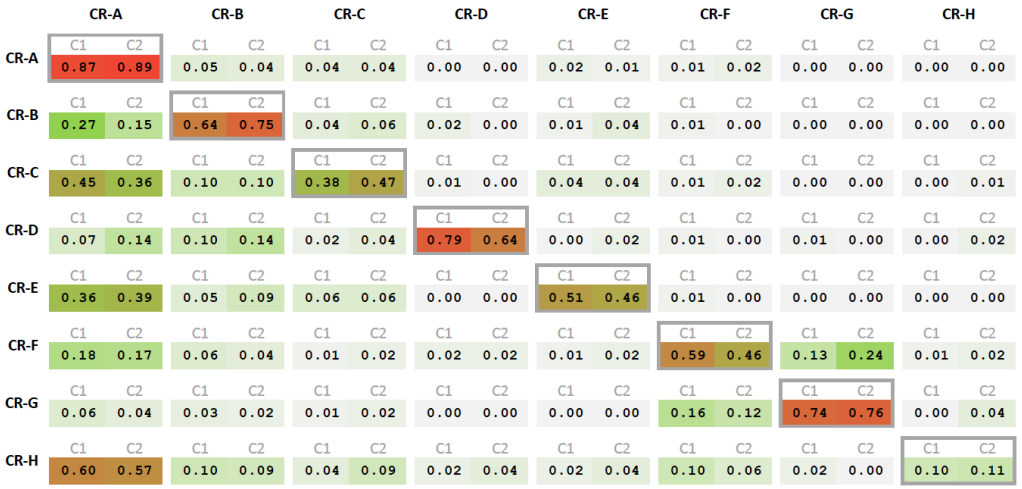

Figure 8: Automated confusion matrix with fractions of total messages per contact reason. Comparison between prediction results on evaluation set with baseline classifier (C1) and the averaged prediction results on the test sets during the baseline cross validation (C2).

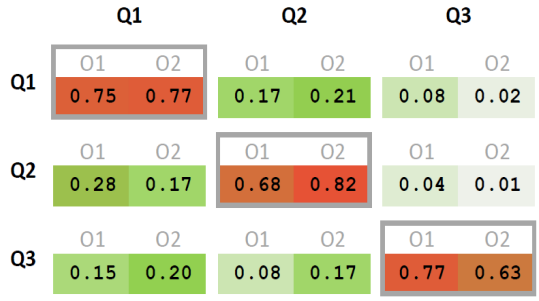

Figure 9: Human confusion matrix with fractions of total messages per queue.

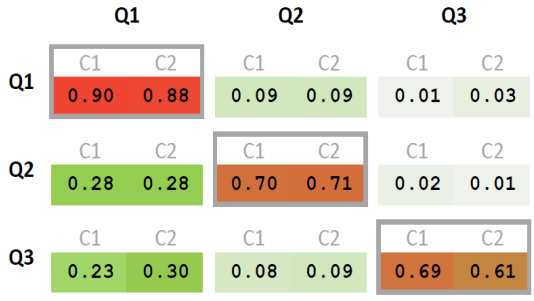

Figure 10: Automated confusion matrix with fractions of total messages per queue.

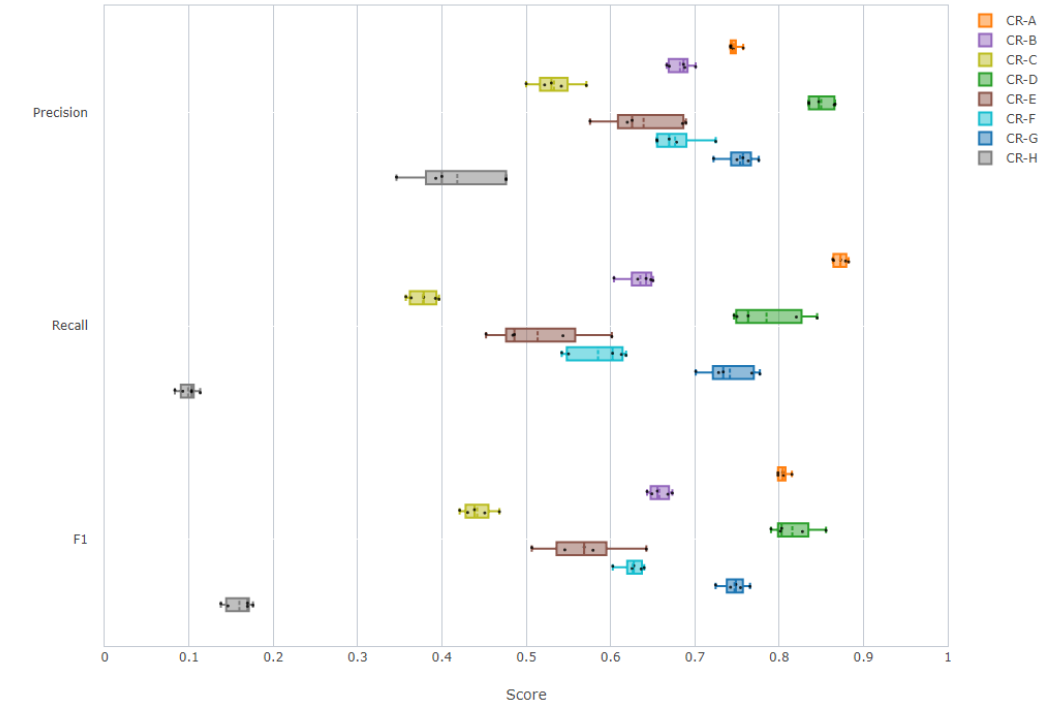

Figure 11: Boxplot of the cross validation performance scores for the baseline model.

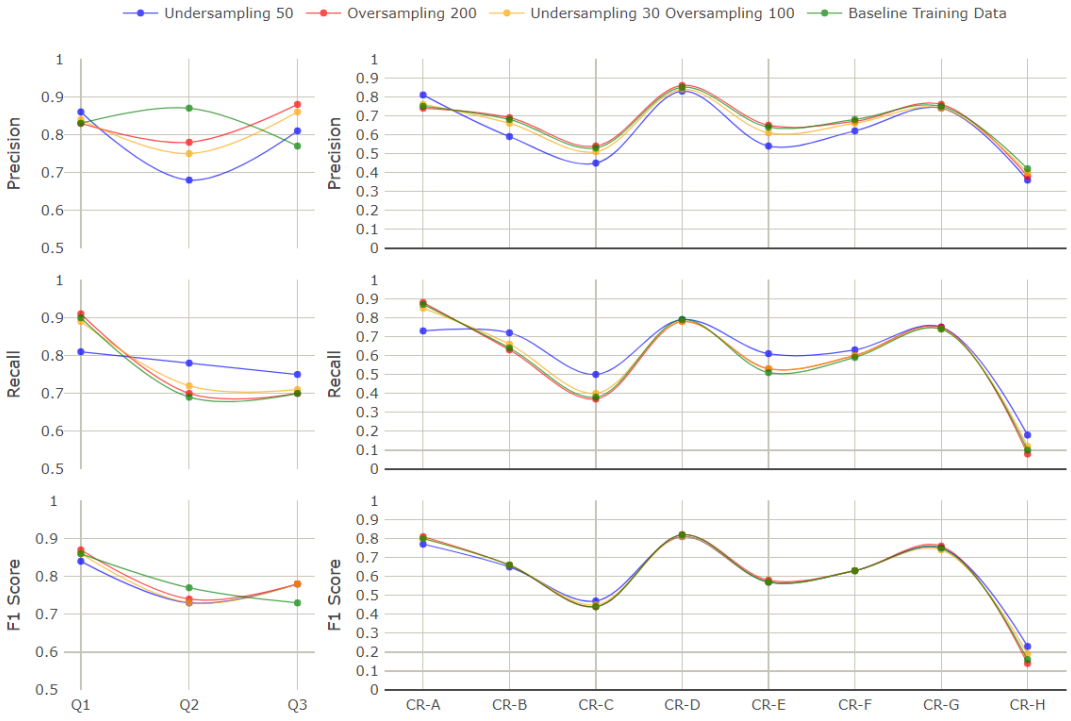

Figure 12: Sampling procedure adjustment comparison with the baseline model.

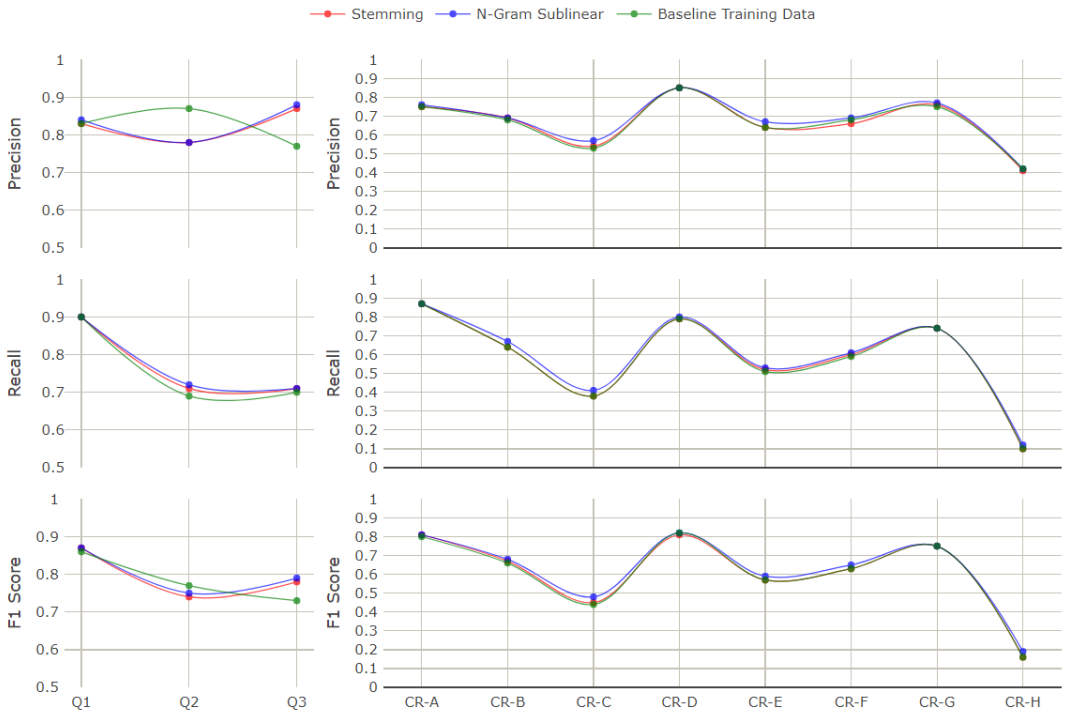

Figure 13: Preprocessing adjustment comparison with the baseline model.

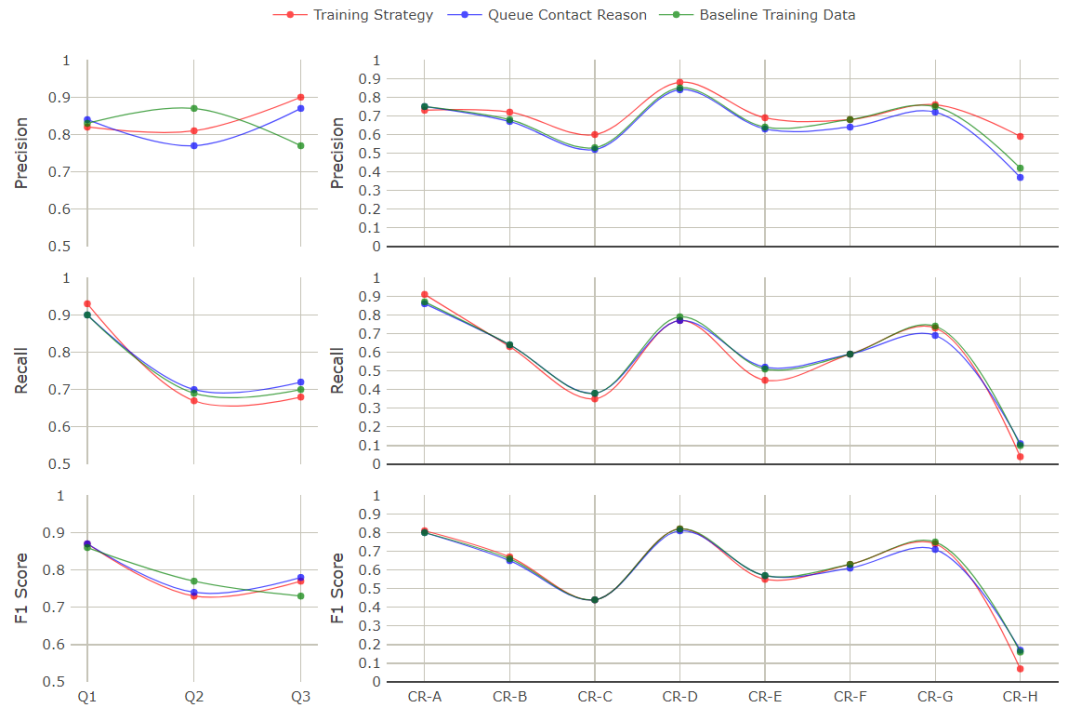

Figure 14: SVM adjustments & ensemble approaches comparison with the baseline model.

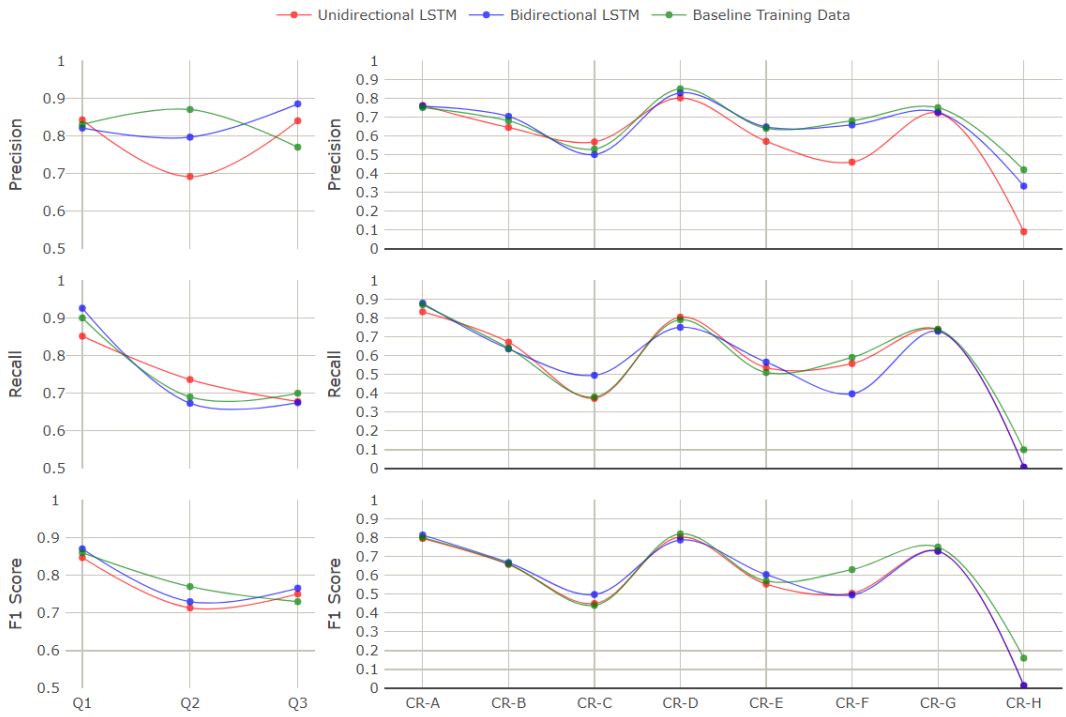

Figure 15: Performance comparison of unidirectional and bidirectional LSTM with baseline model.

Table 7: Improvement results total f1-scores.

Improvement Step	Contact Reason F1	Queue F1
Baseline Training Data	0.70	0.81
Undersampling 50	0.69	0.8
Oversampling 200	0.7	0.81
Undersampling 30 Oversampling 100	0.7	0.81
Stemming	0.71	0.82
N-Gram Sublinear	0.73	0.83
Training Strategy	0.7	0.82
Queue Contact Reason	0.7	0.82
Unidirectional LSTM	0.68	0.8
BiDirectional LSTM	0.7	0.82