Volume 36 (1), pp. 35–71
http://orion.journals.ac.za

ORiON
ISSN 0259–191X (print)
ISSN 2224–0004 (online)
©2020

Sentiment analysis of unstructured customer feedback for a retail bank

J Kazmaier[*] JH van Vuuren[†]

Received: 14 December 2019; Revised: 20 April 2020; Accepted: 23 April 2020

Abstract

With the explosive growth of the Internet and social media, the communication model between an organisation and its customers has become increasingly complex. A problem arises due to the sheer volume of unstructured data that has to be processed for the purposes of studying and addressing customer feedback. This calls for the development of automated methods. Important objectives of such methods include the detection of the underlying sentiment of customer feedback, as well as the synthesis and presentation of this sentiment in meaningful clusters such as topics and geographical locations. In this paper, a case study is conducted in which unstructured customer reviews related to products and services of a South African retail bank are evaluated by means of sentiment analysis. After suitable preprocessing techniques are applied to the reviews, the process of developing suitable models (primarily within the realm of machine learning) for detecting sentiment with a high level of performance is described. Subsequently, model results are analysed, synthesised and visualised in order to extract valuable insight from the data. The findings of the study show that custom learning-based models significantly outperform both pre-trained and commercial tools in sentiment classification. Furthermore, the analysis approach is shown to yield actionable information that may inform decision making.

Key words: Data mining, decision support systems, neural networks, sentiment analysis.

1 Introduction

The opinions of others have influenced the human decision-making process for decades. This is particularly prevalent when a decision involves expending valuable resources such as

[*] Corresponding author: Stellenbosch Unit for Operations Research in Engineering, Department of Industrial Engineering, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: jqkazmaier@gmail.com

[†] **(Fellow of the Operations Research Society of South Africa)** Stellenbosch Unit for Operations Research in Engineering, Department of Industrial Engineering, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: vuuren@sun.ac.za

http://dx.doi.org/10.5784/36-1-668

time and money, in which case people often rely on the past experiences of their peers [8].

With the explosive growth of the Internet and social media, this has become an increasingly observed phenomenon. As the importance of the Internet as a source of information has grown to exceed that of traditional sources of knowledge, it has also become a platform for sharing ideas and experiences. It is now possible to draw on the opinions of a vast pool of people, consisting not only of acquaintances and professional critics, but also of complete strangers posting their opinions in the public domain [40].

One of the consequences of this development for the business sector is the powerful influence that past and current customers now have on each other and on potential future customers. A market study commissioned by *Popimedia*[1] in 2017 indicated that 78% of South African consumers consult online product reviews before making an in-store purchase [48]. According to a *Google* consumer study of over 1 000 participants, 67% of consumers are influenced by such reviews [23]. The collection and study of opinion, using both external data from the Internet and internal data, such as direct customer communication, have therefore become a necessity in many industries [33].

A problem arises, however, due to the sheer volume of data that has to be processed for the purposes of studying and addressing customer feedback. Changes in technology and culture have made it both easier and more common for customers to comment on an organisation's products and services. Furthermore, for every n customers and corresponding company-to-customer communication channels, there are $\binom{n}{2}$ possibilities for customer-to-customer communication. This situation has given rise to the research field of *sentiment analysis* or *opinion mining*, which is concerned with "the computational treatment of opinion, sentiment, and subjectivity in text" [40].

Whereas the task of sentiment parsing is relatively easy for humans, the subtle nuances of natural languages render this task inherently difficult for computers. This is especially true in the South African context, where opinion-bearing expressions may be composed in up to eleven different languages. Furthermore, while there is an abundance of research dedicated to developing algorithms for the purpose of classifying sentiment, little guidance exists on how to incorporate this information into the decision-making processes of affected entities.

In this paper, sentiment analysis methods are applied to extract valuable insight from opinion-bearing text in the context of a case study in the South African banking sector. More specifically, customer reviews on the products and services of a retail bank are analysed in order to determine the typical content of these reviews, the distribution of sentiment expressed by customers, and the relationship between these aspects and additional structured attributes which describe the customer or the branch of the bank he or she had visited prior to submitting the review.

The remainder of this paper is structured as follows. In §2, important concepts pertinent to sentiment analysis are reviewed. The background to the case study is then given in §3, and the results of the case study analysis are presented in detail in §4. Finally, the paper closes in §5 and §6 with a reflection on the study and some proposals for future work, respectively.

[1] Popimedia is a global advertising technology platform and a large Facebook marketing partner.

2 Sentiment analysis of unstructured text

The field of sentiment analysis refers broadly to the study of people's disposition towards certain targets, typically including products or services, public figures, events or current issues. By analysing observations of people's actions in the form of facial expressions, speech or, as in the context of this paper, written compositions, this field aims to extract the "opinions, sentiments and emotions" of the subject [57].

A prevalent task commonly pursued in the realm of sentiment analysis is opinion polarity classification, which aims to classify a text as expressing a *positive* or *negative* opinion. In some cases, a third class, *neutral*, is added [57]. The description of this task, which is the focus of this paper, may be further refined by adopting the following definition of an opinion due to Liu and Zhang [34]:

> An opinion [...] is a quintuple, $(e_i, a_{ij}, o_{ijk\ell}, h_k, t_\ell)$, where e_i is the name of an *entity*, a_{ij} is an *aspect* of e_i, $o_{ijk\ell}$ is the *orientation* of the opinion about aspect a_{ij} of entity e_i, h_k is the *opinion holder*, and t_ℓ is the time when the opinion is expressed by h_k.

If an opinion is expressed on the entity as a whole, the aspect GENERAL is used in place of a, which is henceforth denoted by G. Opinion polarity classification, then, entails classifying the opinion polarity $o_{ijk\ell}$ of a given opinion as either *positive*, *negative* or, if applicable, *neutral* [34].

This analysis can be performed on several levels, namely the *document level*, the *sentence level* or the *aspect level* [34, 36, 57]. At the aspect level, the analysis objective is to find every quintuple $(e_i, a_{ij}, o_{ijk\ell}, h_k, t_\ell)$ in a given document $\boldsymbol{d}$. Opinion polarity classification at the document level, on the other hand, means determining o on aspect G in the quintuple (e, G, o, h, t), given an opinionated document $\boldsymbol{d}$. It is assumed, in this case, that the document is evaluating a single entity e, and that h and t are unknown or irrelevant [34]. This is the level of analysis adopted in this paper. The analysis at sentence level can be viewed as a special case of the document-level analysis, where the document consists of a single sentence or where the opinion polarity of each sentence of a document is determined separately.

2.1 Analysis approaches

Solution approaches to the opinion polarity classification problem may be grouped into two major categories, namely *lexicon-based* approaches and *machine learning* approaches (or statistical approaches) [24, 36, 44]. The former are often also referred to as knowledge-based approaches, since they rely on existing semantic resources such as sentiment lexicons[2] [21, 24]. The latter approaches, on the other hand, are concerned with learning patterns based on labelled, historical data, without the use of any additional resources [21].

The family of lexicon-based approaches is aimed at determining the opinion polarity of a portion of text by counting and weighting the *known* polarity of individual words and

[2] A *sentiment lexicon* provides information on the prior polarity of a word, *i.e.* whether it is considered *positive*, *negative* or *neutral* in most contexts [30].

phrases contained in the text [5, 16, 24]. In the most simple approach, the number of negative words contained in a text is subtracted from the number of positive words. The text is then classified as *positive* if its total score is greater than zero, or *negative* otherwise [16]. Other scoring functions can also be used.

2.1.1 Lexicon-based approaches

Lexicon-based methods may further be differentiated by the way in which the sentiment lexicons are created. There exist three general approaches toward lexicon generation, namely the manual approach, the *dictionary-based* approach and the *corpus*[3]*-based* approach. The former is time-intensive and is therefore typically used in conjunction with the latter two (automatic) methods. More specifically, a small set of seed words is manually labelled with their semantic orientation, and this set is then expanded using automatic methods [24, 36].

Dictionary-based approaches expand the set of seed words based on existing thesauri or other available resources, such as *WordNet*,[4] to find their synonyms and antonyms. Given the fact that words generally have semantic orientations that are of the same orientation as their synonyms, and of opposite orientation to that of their antonyms, new words may be added to the lexicon iteratively, until no new words are found [26]. The advantage of this approach is its simplicity. It has a significant disadvantage, however, in that it cannot recognise the domain-specific polarity or the *contextual polarity* of words [24, 36]. Some words, such as *"warm,"* may generally be considered positive, and therefore have a positive prior polarity. The same word may, however, be used to convey a negative sentiment in other contexts (*e.g. "warm beer"*) [53]. Similarly, a word may carry a positive connotation in one domain, but a negative one in another. A *"long battery life"* for an electronic device, for example, is certainly favourable, whilst a *"long queue"* in the service industry is not.

The problem of contextual polarity is associated with both dictionary-based and corpus-based approaches. Lexicon-based methods are therefore often combined with other, typically rule-based, methods in order to address these shortcomings. On the other hand, the problem of domain-specific polarity can be alleviated by applying corpus-based lexical methods if the documents in the corpus are selected exclusively from the domain in question [24]. According to this approach, the lexicon is expanded based on syntactic or co-occurrence patterns in a large corpus [24, 36].

2.1.2 Machine learning approaches

Opinion polarity classification may be framed as a regular (text) classification problem. More specifically, each document represents a record $i \in \{1, \ldots, n\}$, defined by a set of features $\boldsymbol{x}_i$ and a target class, in this case, its sentiment polarity $y_i \in \{\mathit{positive}, \mathit{negative}, \mathit{neutral}\}$. A set of labelled documents can then be used to train a model to predict the class

[3] A corpus is a collection of several documents.

[4] WordNet is a lexical database for English in which words are organised into synonym sets ('*synsets*'), each representing one underlying concept, and these sets are interlinked by semantic and lexical relations [37].

label for a document instance of an unknown class. Consequently, any existing machine learning algorithm for classification can be applied to this problem [33, 34, 36].

In the literature, machine learning is predominantly applied to the sentiment analysis problem in a supervised setting, in which all training samples are annotated [38]. Approaches also exist for the case where only some samples have labels, as well as for a completely unsupervised setting.

Several literature surveys suggest that most researchers "agree on the learning techniques" of naïve Bayes, *support vector machines* (SVMs) and often also maximum entropy (multi-class logistic regression) [38, 52, 54] for opinion polarity classification. These algorithms were employed by Pang *et al.* [41] in a pioneering paper on machine learning for sentiment analysis and may therefore be referred to as *traditional* methods in the context of sentiment analysis. In recent surveys, deep *artificial neural networks* (ANNs) have also been recognised as an important class of algorithms [38, 57, 58]. This class includes *fully connected* feedforward neural networks (denoted in this paper as ANNs), *convolutional neural networks* (CNNs) and recurrent neural networks, such as the *long short-term memory* (LSTM) network.

In general, machine learning techniques have been shown to outperform lexicon-based methods [1, 9, 24], in some cases with the best performing lexicon-based classifier achieving an accuracy that is approximately 10% lower than that of the worst performing machine learning algorithm and 20% lower than the best performing machine learning algorithm [1]. Dhaoui *et al.* [17] found the performance of both approaches to be similar in terms of the well-known F-measure, although the accuracy achieved by the machine learning approach still proved to be superior.

The performance of machine learning methods depends highly on the model variant, features and data set used [45, 54]. It is therefore advisable to test several configurations on a given data set in order to find one that is most suited for that particular scenario.

2.2 Preprocessing of text

Prior to applying any of the analysis approaches described in §2.1, the unstructured text, which may contain '*noise*' in the form of misspellings, inflected word forms and uninformative words or punctuation marks must be preprocessed or '*cleaned.*' This process typically includes the tokenisation, filtering and normalisation of the text.

Tokenisation refers to the process of splitting documents into their constituent parts. The sentence *"I absolutely loved this movie!"* may, for example, be represented by the set $\mathcal{A} = \{$"*I*", "*absolutely*", "*loved*", "*this*", "*movie*!"$\}$ of individual tokens rather than one long string by segmenting it on the spaces between words.

The resulting sets of tokens are subsequently filtered in order to remove irrelevant or uninformative tokens from the data. This entails, for example, removing punctuation and common *stopwords*, such as *"and," "the,"* and *"it."* The tokens "*I*" and "*this*" would be removed from the above example during this process, and the token "*movie*!" would become *movie.*

Finally, the remaining tokens are normalised to their standard or root form in order to reduce redundancy in the data. This includes many subprocesses, such as the conversion of text to all lower-case, the correction of spelling errors and shorthand, as well as stemming and lemmatisation [43]. The objective of both stemming and lemmatisation is to reduce inflectional or derived forms of a word to a common base form. Whilst stemming is typically a crude process which removes word endings according to some heuristic, lemmatisation makes use of a vocabulary and a morphological analysis[5] of words to return the base form of a word, known as the *lemma* [35]. In the case of the example sentence above, the token *"loved"* would be reduced to its root form, *love*.

2.3 Vectorisation of text

In order to apply machine learning algorithms to the text, it must subsequently be represented in the form of a numerical vector by means of a vectorisation or feature extraction process. There are several possible types of features that may be employed for this purpose, including term-based features, linguistic features and topic-oriented features. The former is the predominant method in the literature, with the bag-of-words model and word embeddings among the most popular text representation models.

2.3.1 The bag-of-words model

After tokenising each of the documents in the corpus, a dictionary of all unique tokens, referred to as *types*, that are contained in the corpus is generated [55, 56]. A matrix may then be constructed in which each document in the corpus constitutes a row, and the columns, or features, are the token types contained in the dictionary. This representation of a text as a set of its words, without regard for possible dependencies between words, is commonly referred to as the *bag-of-words* model [32, 43].

This matrix may be populated in three primary ways. In the Bernoulli document model, each table entry a_{ij} represents the presence or absence of a token of type j in document i. Each document is thus represented by a feature vector with binary elements taking the value 1 if the corresponding term in the dictionary is present in the document, or 0 otherwise [5].

Instead of employing the presence-based Bernoulli model, it is common in natural language processing applications to represent a text by means of a frequency-based feature vector [57]. In the multinomial document model, for example, each entry of the feature vector is an integer value corresponding to the frequency count of that term in the document [5]. Alternatively, a frequency weight may be used, which takes into account the frequency of a term in a document relative to the frequency of the same term in the entire corpus.

One such popular frequency weight is the *term frequency-inverse document frequency* (TF-IDF) weight [40, 43]. Many variants of this weighting scheme exist, but it is typically composed of two terms: The *term frequency* (TF) and the *inverse document frequency*

[5] In linguistics, morphology is the study of words, their formation and internal structure [2].

(IDF). The former is calculated as

$$T(t, \boldsymbol{d}) = \frac{f_{t,\boldsymbol{d}}}{\sum\limits_{t' \in \boldsymbol{d}} f_{t',\boldsymbol{d}}},$$

where $f_{t,\boldsymbol{d}}$ is the frequency count of term t in document $\boldsymbol{d}$ and the denominator represents the total number of terms in the document. By normalising the frequency count over the length of the document, a bias toward longer documents, in which frequency counts of a given term tend to be higher, is avoided. The inverse document frequency is a measure of the information provided by a term. The assumption is that terms which occur frequently across documents, such as "the," "are" and "is," provide little meaningful information and should receive a lesser weight than terms which occur more rarely. It is calculated as

$$I(t, \boldsymbol{C}) = \log \frac{N}{n_t},$$

where N is the total number of documents in the corpus $\boldsymbol{C}$ and n_t is the number of documents in the corpus which contain the term t. The value is therefore zero if the term is contained in all documents of the corpus, and increases with an increase in the rarity of the term. The frequency weight $F(t, \boldsymbol{d})$ is finally calculated as the product of $T(t, \boldsymbol{d})$ and $I(t, \boldsymbol{C})$ [43].

In order to capture a wider context, one may also employ higher order *n-grams*, rather than individual terms (*unigrams*) as document features. These features represent groups of adjacent words in their original word order [57]. It stands to reason that a better performance for classifying sentiment can be achieved by viewing, for instance, negated phrases such as "not happy" as a single attribute. There is, however, controversy in the literature in this regard. Whilst Pang *et al.* [41] found that unigrams outperformed bigrams in classifying the sentiment polarity of movie reviews, Dave *et al.* [15] reported bigrams and trigrams to yield better results in product review classification [40]. It has, in fact, often been found that the problem of sentiment classification, and therefore also the effectiveness of certain feature sets, is domain-specific and context-sensitive [10].

2.3.2 Word embeddings

When tokens contained in the dictionary are used as features to represent text, it may appear as if the feature space becomes excessively or impracticably large. Often, this concern is unwarranted, since most documents only feature a small subset of the words contained in the dictionary, resulting in a sparse matrix (in which most elements are zero). Models operating on the feature space may therefore leverage this property by storing only positive values [43, 56].

On the other hand, these sparse vector representations have significant disadvantages. First, statistical models are more difficult to train with such data. These models are then prone to overfitting and require larger sample sizes in order to train effectively [6, 50]. Secondly, representing text data in this way provides no meaningful information to the model as to the relationships that may exist between words.

Word embeddings aim to overcome these problems by representing each word not as an index in the vocabulary, but as a fixed-length numerical vector [58]. The size of this vector can be chosen so that the size of the matrix is reduced, resulting in lower computational cost for models employing these data as input. Furthermore, these representations are dense, allowing similar words to be placed near one another in the transformed vector space.

Such representations may be generated by means of linear transformations of the sparse vector space, for example by applying *principal component analysis* (PCA) or latent semantic analysis, or through the use of algorithms which can *learn* useful data representations [42, 58]. A word embedding layer may also be trained end-to-end with a neural network [4].

3 Background to the case study

The industry partner associated with this study is a South African retail bank with over six million customers[6]. In an attempt to monitor customer satisfaction levels, the bank launched an initiative to elicit customer feedback *via* SMS in association with a third-party vendor specialising in collecting and analysing customer experience (*voice of the customer*) data. The process by which the communication with customers is carried out is as follows. After visiting a branch, randomly selected customers are sent an SMS requesting them to rate their experiences on a scale of 1 (*great*) to 3 (*bad*). Customers who respond with a negative rating (2 or 3) are then sent a follow-up SMS asking them to elaborate on their negative experiences and provide reasons for their rating (if the rating was 3), or suggestions as to how the bank may improve its service (if the given rating was 2). There is no follow-up on positive ratings in an effort to reduce costs.

The third-party vendor subsequently analyses the free-form customer responses to the follow-up messages using proprietary sentiment analysis software in order to assign sentiment scores of -1 (*negative*), 0 (*neutral*) or $+1$ (*positive*) to these responses. After analysing a batch of messages, it turned out that several customers had misunderstood the rating scale, assuming that a higher number translated to a positive rating, rather than a negative one. Therefore, although there had only been follow-up in respect of neutral or negative ratings, not all sentiment subsequently expressed in the messages was necessarily negative.

The bank is presently experiencing two primary problems with its current approach to customer satisfaction monitoring. First, upon investigation by the data science department of the bank, the third-party software used to analyse the sentiment expressed in the customer feedback messages was not deemed sufficiently accurate in classifying sentiment polarities. Secondly, after messages with negative sentiment are identified, employees have to examine these messages manually in order to gain actionable insight. Applying sentiment analysis may alleviate both of these problems by aiding the data science department in building and testing more accurate models for classifying sentiment tailored to the specific context, as well as by facilitating an in-depth analysis of the content of the messages

[6] The anonymity of the industry partner is protected by a non-disclosure agreement.

in each sentiment category and the relationship between additional customer data and customer satisfaction without the need for manually reading each review.

3.1 Data preparation

The data employed in this case study were received in various separate data sets. These data sets were transformed into two relational data tables, namely *reviews* and *supplementary data*. The *reviews* data set contains the attributes related to the customer's branch visit, including the name, type, town, province and geographical coordinates of the branch, as well as the primary need associated with the branch visit and the type of consultant that attended to the customer. Furthermore, this data set contains the initial response of the customer to the SMS (*i.e.* the rating of 1, 2 or 3 known as the *Q01 Value*), as well as the free-form text response given as a result of the follow-up SMS.

The supplementary data set contains attributes further describing the customer, including demographic information, such as age and gender, as well as information related to the customer's bank account, such as their current loan status and average monthly bank fee.

3.2 Annotating customer responses

As was discovered by the data science department of the industry partner, the third-party software did not seem to yield sufficiently accurate sentiment polarity classifications of the messages sent in by customers. In respect of a sample of 500 entries that were manually labelled by an employee of the data science department at the industry partner, the human annotator and the software were in agreement in only 60.4% of the cases. In order to formally evaluate the accuracy achieved by the software, and in order to evaluate and compare the newly developed models, a *ground truth* data set was required.

Establishing such a data set is a particularly challenging task in the case of sentiment analysis since, in contrast to other classification tasks, human annotators typically agree on a sentiment label only 80% of the time [12, 28, 39]. An attempt was, however, made to reduce the uncertainty associated with the assigned labels. A group of annotators was gathered from within the Department of Industrial Engineering at Stellenbosch University, representing diverse cultural backgrounds, an age range of approximately 20 years and both genders. A total of 2 500 reviews were then randomly selected from the data to serve as the ground truth data set. Each review was labelled as *positive*, *negative* or *neutral* independently by at least two human annotators. If these annotators agreed on a sentiment class, this class was assigned as the *true* class label. If not, a third annotator was asked to classify the review. If this annotator agreed with one of the first two annotators' classifications, then that class was assigned as the true class label. If not, additional annotators were asked to assess the review until a majority vote was achieved for one of the classes.

After the first round of annotations (with two annotators per review), the agreement between annotators was 81.8%, which is consistent with the literature. Furthermore, 11.8% of the labels assigned to reviews (two per review) were *positive*, 69.5% were *negative*, 15.1% were *neutral* and 3.6% were marked as *uncertain*. After the second round of annotation (where additional annotators were added in cases of disagreement), a label was found for each of the 2500 reviews with a final inter-annotator agreement of 93.92%. A total of 283 (11.3%) of the reviews were classified as *positive*, 1 819 (72.8%) were classified as *negative* and 398 (15.9%) were classified as *neutral*.

4 Case study results

In order to extract valuable information from the case study data, a three-stage process was followed. First, the data were preprocessed according to the tokenisation, filtering and normalisation procedures described in §2.2. Subsequently, a structured model building and evaluation process was applied in order to find a suitable model able to classify the documents accurately as either *positive*, *negative* or *neutral* in sentiment, employing several of the analysis approaches described in §2.1 in conjunction with the vectorisation techniques described in §2.3. This step was carried out in respect of the labelled subset of the data. The most suitable model was then deployed to the remaining unlabelled reviews, yielding a distribution of sentiment across the corpus. In order to extract valuable information and insight from these results, a subsequent analysis of the results was performed. The objective of this analysis was to identify patterns in the data that may indicate trends in and possible causes of the sentiment expressed by the bank's customers.

4.1 Preprocessing and cleaning

The original corpus of customer reviews contained 12 448 unique tokens (*types*) between the 10 354 documents. A *word cloud*[7] of the unprocessed corpus is shown in Figure 1. From this figure, it is evident that stop words such as *and* and *it* are the most frequent words in the corpus. In order to determine the effect of various preprocessing steps on the number of unique tokens, only the tokenisation step was applied at first. Various other preprocessing operations were then added to the selection iteratively, evaluating the effects of the sequence of operations in each case.

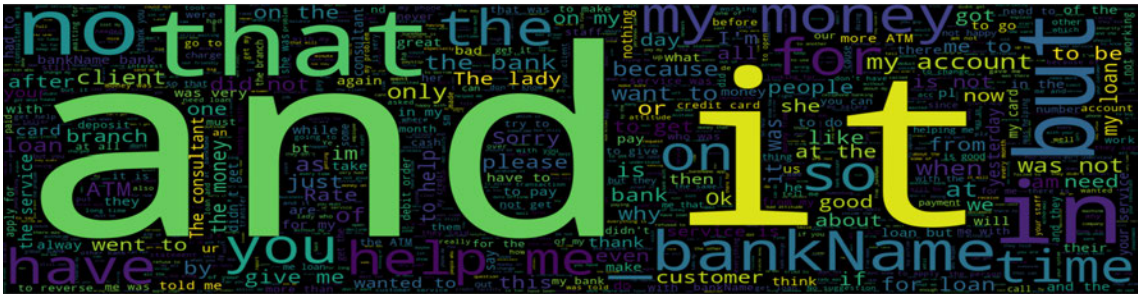


Figure 1: *A word cloud illustrating the most frequent tokens in the original, unprocessed corpus.*

[7] A word cloud is a visualisation of the most frequently occurring terms in a corpus. Term frequencies are indicated by the relative sizes of the words, where frequent words are shown in larger font sizes (the word colouring is arbitrary).

A summary of these effects, expressed as the total number of types in the corpus, is given in Table 1. From the table, it is clear that 344 (12 448 − 12 104) unique tokens were removed from the corpus during the removal of stopwords from `Python`'s *Natural Language Toolkit* (`NLTK`) library. Although the list itself consists of only 175 words, the words are removed without regard for case (whilst the tokens "*It*" and "*it*" are considered to be two distinct types, both words would be removed from the corpus based on the stop word "*it*"). A further 26 (12 104 − 12 078) types were removed from the corpus with the removal of punctuation marks. Since the `NLTK` stop word list is not tailored to a sentiment analysis problem, however, certain words were excluded from this list. These include the words *but, no, nor, not, don't, aren't, couldn't, didn't, doesn't, hadn't, hasn't, haven't, isn't, wasn't, weren't, wouldn't,* and *won't*, which could indicate shifts in sentiment, as well as the terms *above* and *below*, which could also be used to indicate sentiment (*e.g. "the service was below standard"*). Finally, the intensifiers *too* and *very*, and the exclamation mark were also excluded from the list. With the amended list, 24 (12 102 − 12 078) of the types that were removed in preprocessing Sequence 3 in Table 1 were preserved in Sequence 4.

Sequence number	Processing steps applied	Number of types
1	Tokenisation	12 448
2	1 & stop word removal	12 104
3	2 & punctuation removal	12 078
4	3 with an amended stop word list	12 102
5	4 & grouping of numbers	11 497
6	5 with certain numbers excluded	11 500
7	6 with case correction	9 404
8	7 with spelling correction	5 385
9	8 & Porter stemming	3 951
10	8 & Lancaster stemming	3 484
11	8 & Snowball stemming	3 920
12	8 & Lemmatisation with WordNet	4 948

Table 1: *The effects of various preprocessing steps on the size of the vocabulary.*

In Sequences 5 and 6, uninformative numbers were removed from the data set as part of the filtering process. These include reviewer's phone numbers or amounts of currency, which are rarely informative, since any model would identify these as distinct features and therefore not be able to *learn from experience*. Such numbers may be grouped into a single feature indicating the presence of some numerical value. More specifically, if a token contained more numerical than non-numerical characters, it was treated as a number and replaced by the generic token *_num*. This distinction allows currency values such as "R100" to be treated as numbers whilst shorthand notations and typos such as "*gr8*" or "*2yes*" are treated as text.

The grouping of tokens with primarily numerical characters into a single token further reduced the size of the vocabulary by 605 types to 11 497. Since customers were asked to rate the bank on a scale from 1–3 in the original feedback request, however, the numbers 1, 2 and 3 were excluded from this aggregation. Overall, the filtering of the corpus (Sequences 1–6) reduced the size of the resulting vocabulary by 7.6% ((12 448 − 11 500)/12 448). The subsequent application of case normalisation had a much greater effect, reducing the number of types by 2 096 (11 500 − 9 404).

In the following sequence (Sequence 7), a spell correction algorithm was applied, which makes use of the popular *Aspell* [3] spell checking library. If incorrect spelling is detected, a list of suggestions to correct the spelling error are extracted using Aspell. Subsequently, a decision must be made as to which suggestion, if any, should be accepted to replace the current token. In order to determine this, a simple check was first performed to determine whether the misspelt word is the same as one of the suggestions when case was disregarded. If the word "*amazing*," for example, were accidentally typed as "*amAzing*," the suggestion "*amazing*" would immediately be accepted. If this was not the case, the frequencies of the current, apparently misspelt token and each of the suggestions in the corpus were retrieved. The token that was most frequently used in the corpus was then selected as the correct spelling[8]. This allowed the spell checking algorithm to correct the spelling of a word with consideration of the context in which it was used. If, for example, names or jargon are frequently used in the corpus that are flagged as incorrect by Aspell (*e.g. "Stellenbosch"* or "*hyperparameters*"), such words are not corrected. The application of this algorithm further reduced the vocabulary size by 4 019 (9 404 − 5 385), amounting to over 30% of the original vocabulary size. This is testament to the fact that the data were ridden with spelling errors.

Finally, various different stemming or lemmatisation algorithms were applied to the filtered, case-normalised and spell-corrected corpus in Sequences 9–12. The *Lancaster stemming algorithm* [25] appeared to be the most *'aggressive'* stemming algorithm, followed by the *Snowball stemming algorithm* [47] and *Porter's stemming algorithm* [49] with these algorithms yielding final vocabulary sizes of 3 484, 3 920 and 3 951, respectively. This effect is significantly smaller for the lemmatisation algorithm built upon WordNet, which reduced the vocabulary to a final size of 4 948. In spite of this reduced normalisation effect, Sequence 12 was chosen as the final preprocessing sequence for the data, in view of the fact that one of the objectives of the case study was to determine which types of models perform better in respect of the particular data set. Since the lexicon-based models applied during the modelling stage make use of pre-compiled sentiment lexicons, it is necessary that the preprocessed documents comprise words which are contained in the English dictionary. The word cloud of the resulting, preprocessed corpus is shown in Figure 2.

Apart from the retained stop words *no*, *not* and *but*, and the aggregated token *_num*, words typically associated with a financial bank constitute the most frequently observed words in the corpus. These words include *bank, loan, money, account, service, branch, client* and the name of the bank representing the industry partner (which was replaced by the token *_bankname* in the interest of anonymity). Terms which stand out more from the context are *help*, *problem* and *time*, which may warrant further investigation during the analysis stage. Overall, the applied preprocessing operations reduced the vocabulary of the corpus by over 60% ((12 488 − 4 948)/12 488) *via* filtering and normalisation, bringing information-bearing words to the forefront.

[8] The token appearing in the original review is retained in the case of a tie.

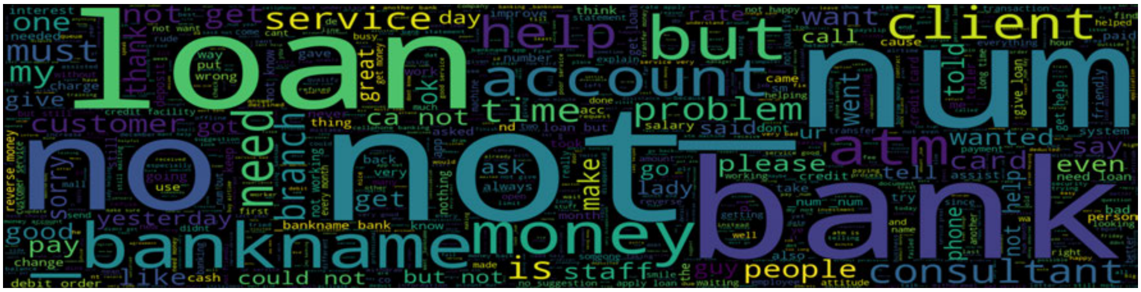


Figure 2: *A word cloud illustrating the most frequent tokens in the final, preprocessed corpus. The name of the industry partner was replaced by the token "_bankname" in order to preserve anonymity.*

4.2 Model development

In order to find a suitable model for the sentiment expressed in customer reviews, several lexicon-based and machine learning algorithms were developed and evaluated in respect of the annotated ground truth data set. Four *off-the-shelf* lexicon-based models were also employed for benchmarking purposes, namely *Sentiwordnet*, *Pattern*, *Vader* and the *Hu and Liu Opinion Lexicon*. Each of these models is already equipped with a sentiment lexicon and could therefore readily be applied to input data.

Sentiwordnet [18] is a lexical resource in which each synset in the WordNet database (see §2.1) is assigned an objectivity score, a positivity score and a negativity score. In this case study, the polarity score of a token was computed by subtracting the negativity score from the positivity score of the synset to which the token most likely belongs[9]. The scores for all tokens in a document are then summed to yield a total score for the document. As per the recommendation in the documentation of the `Pattern` library [14], which was used to implement the Sentiwordnet model, a document is classified as *negative* if the total score is negative, as *positive* if the total score is greater than 0.1 and as *neutral* otherwise. The same rule was applied to determine a document's sentiment class based on its polarity score according to the Pattern model, which refers to the built-in sentiment model of the `Pattern` library. This model takes as input the entire document; therefore, an aggregation of individual token scores is not necessary.

Vader [27] (an acronym for *Valence Aware Dictionary for sEntiment Reasoning*) is a rule-based model which assigns a sentiment score to documents based on a lexicon of 7 500 empirically evaluated features commonly occurring in social media blogs, as well as generalisable heuristics used by humans. Documents are classified as *positive*, *negative* or *neutral* based on the sign of this score, where the neutral class is assigned for a score of zero.

Finally, the opinion lexicon developed by Hu and Liu [26] contains a list of adjectives along with their sentiment polarities (*positive*, *negative* or *neutral*). This lexicon is employed in conjunction with a simple counting rule: Documents containing more positive than

[9] A word can belong to several WordNet synsets, which are distinguished by their contextual meaning. By selecting the first synset suggested by the database, complicated word sense disambiguation procedures were avoided.

negative adjectives in the lexicon are classified as *positive*, documents with an equal number of positive and negative adjectives are classified as *neutral*, and the remaining documents are classified as *negative*.

These lexicon-based models were compared with six machine learning algorithms for sentiment classification, trained by the authors in respect of the case study data. These included the three *traditional* algorithms for sentiment classification mentioned in §2.1, namely naïve Bayes, SVMs and maximum entropy, as well as three different neural network architectures, namely an ANN, a CNN and an LSTM network. Due to the sequential nature of the latter two algorithms, documents were represented by means of a word embedding layer trained end-to-end with the network for these algorithms. For the remaining algorithms, several variants of the bag-of-words model were used as input concurrently, and the best performing model-feature combination was selected. More specifically, the Bernoulli document model, multinomial document model and the TF-IDF weighting method were used to vectorise the documents, each of which were applied to bag-of-words representations using unigrams, bigrams, or unigrams and bigrams. This resulted in nine (3×3) feature sets per algorithm. In terms of feature selection, the k most frequently used n-grams were selected.

For each model-feature combination (henceforth referred to as an experiment), the vocabulary size k, as well as the hyperparameters of the machine learning algorithm had to be selected prior to training. These included, for example, the tuning parameter C, which controls the bias-variance trade-off in both the SVM and maximum entropy, and the number of layers and learning rate for the ANN.

The hyperparameter tuning process was facilitated by means of a grid search — one of the most common approaches to hyperparameter tuning [13]. According to this approach, a list of possible values is constructed for each hyperparameter. Every possible combination of values is then tested, and the one achieving the highest performance in respect of the validation data is selected. Since the number of combinations that have to be evaluated by the grid search algorithm can quickly grow large, however, it is desirable to limit the number of values tested for certain hyperparameters or features during each iteration. In the remainder of this section, the process followed in pursuit of this objective is first described. Subsequently, the results of the models generated by this limited grid search are presented and discussed.

Empirically, two factors have a large influence on the computational time of a grid search. First, the metrics used to evaluate the performance of a particular model incur a varying computational cost. Secondly, if the size of the vocabulary (the number of tokens or features used) grows, the time required to train the machine learning algorithms in respect of the term-document-matrix can become excessively long. The approach adopted during the case study therefore sought to limit the computational burden of each iteration by selecting the computationally most efficient metric and the smallest possible vocabulary size without significantly compromising on the performance achieved by the models.

For this purpose the smallest, computationally least expensive model, naïve Bayes, was evaluated for varying values of the vocabulary size employing the unigram presence vectorisation as input. More specifically, the model's only hyperparameter, the smoothing parameter α, was selected as one of the values in the set $\{0.0001, 0.2, 0.4, 0.6, 0.8, 1\}$ by means of a 3-fold cross-validated[10] grid search in respect of 80% of the data. Each grid search was performed twice — once using *accuracy* (measured as the proportion of correctly classified observations) and once using the *AUC score*[11] (the area under the *receiver operating characteristics* curve, see [19]) to evaluate performance in respect of each of the three folds. Whilst the AUC score is the preferred metric due to its invariance to class imbalance, the calculation of this metric is more expensive since it must be computed for each of the three classes separately and then averaged, as opposed to the accuracy metric, which is computed directly. As a measure of reference, performing one iteration of the grid search (fitting and evaluating three folds) according to the naïve Bayes algorithm for a vocabulary size of 50 took between 0.1 and 0.4 seconds when using the accuracy metric, and between 0.4 and 0.9 seconds[12] when using the AUC metric, depending on the document model and n-gram range. For larger vocabulary sizes and more complex learning algorithms, which require several iterations of a gradient-based optimisation algorithm, this makes a significant difference.

The results of these experiments are illustrated in Figure 3, where each data point represents the AUC score achieved in respect of the test data (20% of the total data), both in the case where the grid search was performed to maximise AUC score and in the case where the grid search was performed to maximise model accuracy. The training and test data were fixed for these experiments, since the objective was to estimate the *relative* effect of the vocabulary size on model performance.

As is evident from the figure, the AUC scores achieved by the models vary very little as a function of the metric used during cross-validation. Furthermore, increasing the size of the vocabulary has a significant positive effect on performance between 20 and 200 tokens, but this effect reaches a plateau after a vocabulary size of approximately 200. Based on these results, the accuracy metric was employed for all subsequent grid searches. Moreover, a vocabulary size of 250 was provisionally selected for the case study analysis.

Due to the large number of hyperparameter combinations that could be formed for the three deep learning models (ANN, CNN and LSTM), a manual hyperparameter tuning approach was employed to identify good hyperparameter ranges prior to the grid search. During this process, the chosen vocabulary size of 250 was implemented, along with a simple unigram presence document representation for the ANN algorithm. A similar process was followed for all three algorithms. More specifically, a small, simple network was employed as a starting point. The graphs of the training and validation losses and accuracies were then scrutinised *via* a `Tensorboard`[13] interface in order to inform the necessary changes that should be made to the model. This process was repeated until a satisfactory outcome was observed (in terms of the loss and accuracy curves). In each

[10] During *k-fold cross validation*, the training data are partitioned into k subsets, or folds, and k training iterations are performed. During each iteration, the model is trained on $k-1$ folds and the remaining fold is used as a validation set.

[11] In order to apply this binary metric to the ternary problem encountered in the case study, the AUC score was calculated for each sentiment class according to a one-versus-all approach and then micro-averaged according to the number of observations in each class. For discrete classifiers, the ROC curve was approximated.

[12] These values were obtained using a 2018 Mac Book Air with a 1.6 GHz Dual-Core Intel Core i5 processor and 8GB of random access memory.

[13] `Tensorboard` is a well-established framework for visualising neural networks [51].

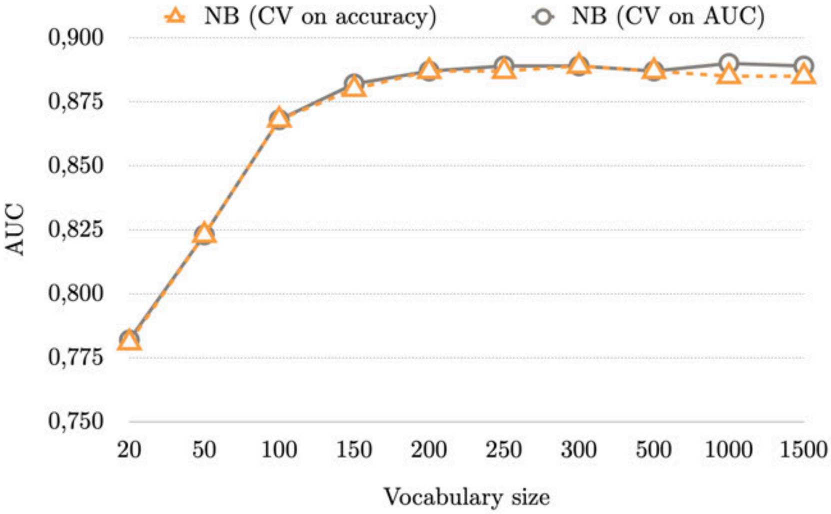


Figure 3: *The effect of vocabulary size on the AUC value achieved by a unigram presence naïve Bayes model in respect of a fixed test data set.*

case, the training data constituted 70% (1 750 observations) of the total labelled data and the validation data constituted 10% (250 observations) of the total labelled data.

For example, the starting point for the ANN algorithm was a simple network with one hidden layer comprising ten neurons. The widely used *rectified linear unit* (ReLU) [31] activation function was applied to these neurons. Initially, no regularisation or batch normalisation procedures were applied. Furthermore, the popular ADAM optimisation algorithm [29] was employed with an initial learning rate of 0.2, an initial learning rate decay of zero and an initial training duration of ten epochs. Examining Figure 4, it may be concluded that the network is indeed *learning* during training, since the binary cross-entropy training loss decreases with the number of epochs. The validation loss, however, diverges as the number of training epochs increases. This indicates that the model may be overfitting the data. In an attempt to curb this effect, ℓ_2 regularisation [22] was applied to the network with a small regularisation parameter value of $\lambda = 0.001$. Furthermore, since there was no evidence of a stagnation in the training accuracy after ten epochs, the number of training epochs was increased to twenty.

These changes to the network structure had the desired effect on the graphs in Figure 4. The validation loss now followed the decreasing trend of the training loss. The new loss curve, however, exhibited a shallower decrease, indicating that the learning rate may be too large. The learning rate was therefore decreased during the next iteration. The remaining hyperparameters were selected in a similar fashion for each of the deep learning algorithms.

Having established suitable hyperparameter values for the three deep learning algorithms, the selection of the vocabulary size based on the naïve Bayes classifier could be verified using the other algorithms. More specifically, six of the ten vocabulary sizes tested in Figure 3 were employed to evaluate the remaining machine learning models. As before,

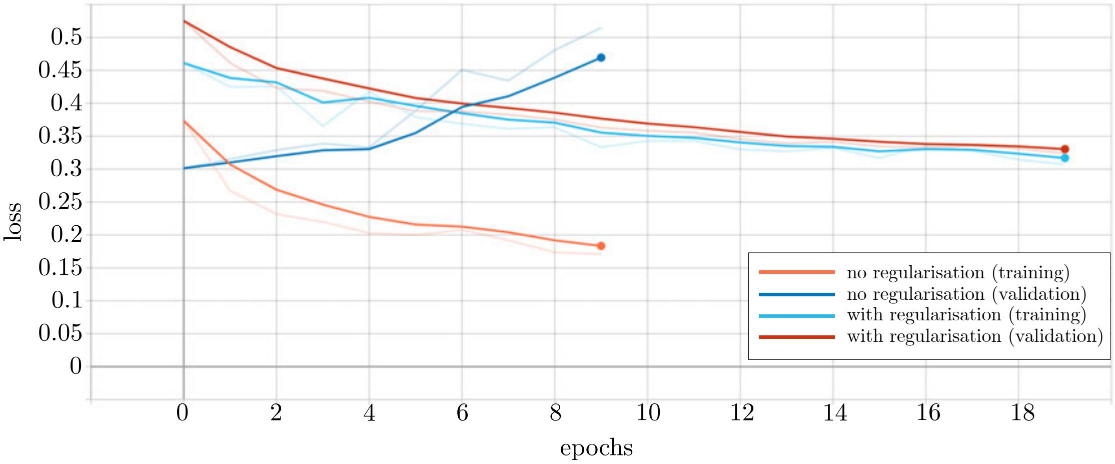


Figure 4: *Tensorboard graphs for the ANN with and without regularisation.*

a grid search with 3-fold cross-validation was performed to tune the hyperparameters of the models. The hyperparameter values included in the grid search for each algorithm are shown in Table 2. For the SVM, logistic regression and ANN algorithms, the unigram presence vectorisation method was employed.

The results in Figure 5 reflect the AUC score in respect of a fixed test set as a function of the vocabulary size. In each case, 3-fold cross-validation was performed employing the hyperparameter ranges given in Table 2 and selecting the combination yielding the best accuracy, as per the observation in Figure 3 that employing this metric during the grid search, while more computationally efficient, results in an AUC score similar to that achieved using the AUC metric. As may be seen in the figure, the AUC scores for all models initially increase significantly with the size of the vocabulary, but the scores then remain relatively constant from a size of approximately 150–200 tokens onwards. These results verify those returned by the experiments in Figure 3. A vocabulary size of 250 was therefore retained for all subsequent experiments.

Finally, given the reduced selection of hyperparameters and a suitable vocabulary size, the grid search could be performed and the resulting models evaluated and compared. As described above, nine variants of the bag-of-words model were implemented, along with the hyperparameter values given in Table 2. In this manner, the effect of each document model could be evaluated, as well as the effect of using longer n-grams both instead of and in combination with shorter n-grams. Consequently, nine experiments were performed for the naïve Bayes, SVM, logistic regression and ANN algorithms, whilst one experiment was performed for the CNN, LSTM, Vader, Pattern, Sentiwordnet and Hu and Liu opinion lexicon algorithms. The total number of experiments was therefore forty two $((4 \times 9) + 6)$. In order to account for the variability of the performance of all the models originating from the training data and test data, as well as the variability of the deep learning models originating from the random initialisation of network weights, each experiment was repeated ten times with a different random seed for each replication. The results are therefore presented in the form of box plots, indicating the median performance, as well as the degree of variation around this median.

Algorithm	Hyperparameter	Values
Naïve Bayes	α	0.0001, 0.2, 0.4, 0.6, 0.8, 1
SVM	C	0.1,1,10
	γ	0.01, 0.1, 1
	Kernel	linear, radial, sigmoid, P2
Logistic regression	C	0.01, 0.1, 1, 10
	Optimiser	SAG, Newton-CG, LBFGS
	Max iterations	100
ANN	Neurons per hidden layer	10,10
	Activation function	ReLU
	Regularisation	ℓ_2
	λ	0.001
	Dropout probability	0
	Batch normalisation	No
	Loss Function	Cross-entropy
	Optimiser	ADAM
	Number of epochs	10, 12
	Initial learning rate	0.02
	Learning rate decay	0
CNN	Embedding size	10
	(Kernel size, stride, number of filters)	(1 1 20)
	Convolution type	Valid
	Pooling	No
	Activation function	ReLU
	Regularisation	ℓ_2
	λ	0.01
	Batch normalisation	No
	Loss Function	Cross-entropy
	Optimiser	ADAM
	Number of epochs	10, 12
	Initial learning rate	0.01
	Learning rate decay	0
LSTM	Embedding size	10
	LSTM output size	10
	Dropout probability	0
	Regularisation	None
	Loss Function	Cross-entropy
	Optimiser	ADAM
	Number of epochs	4, 8
	Initial learning rate	0.01
	Learning rate decay	0

Table 2: *The hyperparameter values tested during the grid search for all machine learning algorithms.*

The AUC scores achieved by each algorithm in respect of the test set (comprising 20% of the data) after hyperparameter tuning are shown in Figure 6. Where multiple models were trained by means of the same algorithm (*i.e.* naïve Bayes, logistic regression, SVMs and ANNs, with the nine variations of the term-document matrix taken as input), the best-performing model was selected for each of the ten replications. It is clear from the figure that the machine learning models outperformed the off-the-shelf lexicon-based models by a large margin. Whilst the four lexicon-based models achieved similar median AUC scores in the region 0.6–0.62, constituting performance only marginally better than

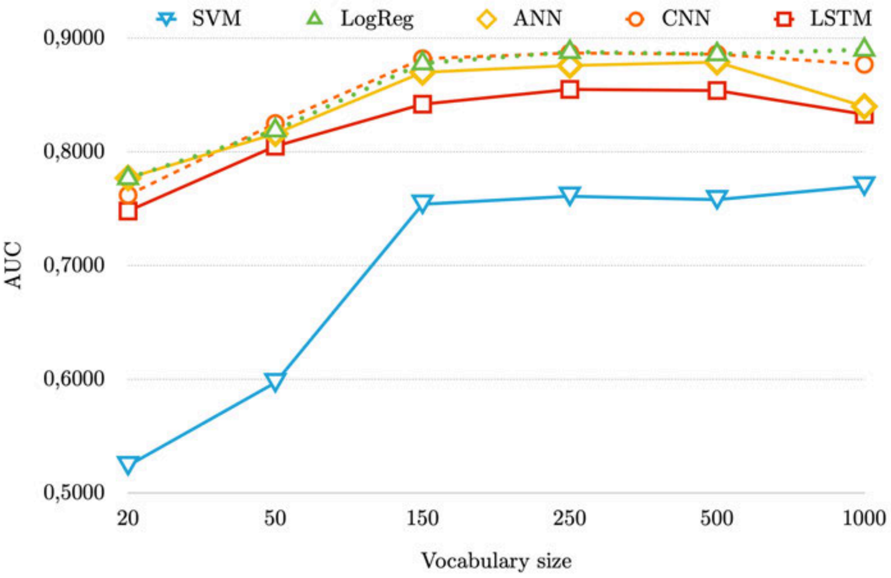


Figure 5: *The effect of vocabulary size on the AUC value achieved by various machine learning models.*

random guessing, the machine learning models achieved scores in the range of 0.79–0.91, with five of the six models achieving median AUC scores of over 0.89.

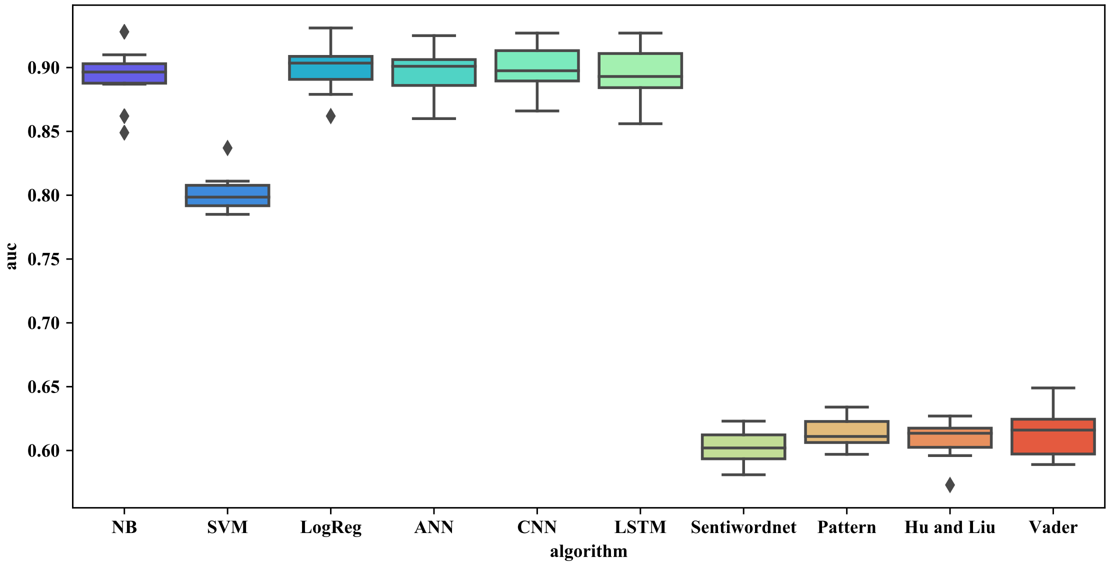


Figure 6: *Model performance in terms of AUC. The AUC values achieved during ten experimental runs are shown in the form of box plots.*

It would appear that the SVM is significantly outperformed by the other models in respect of the AUC score. It is important to consider, however, that the SVM is the only discrete[14]

[14] A discrete classifier assigns observations to classes directly, in contrast with a probabilistic classifier,

classifier among the machine learning models. The AUC score is therefore merely an estimate, based on the linearly interpolated approximation of the ROC graph between three single points (the origin, the classifier's true positive and true negative rate and the point $(0, 1)$). Hence it is possible that this score misrepresents the actual performance of the SVM algorithm to some degree. Whilst the lexicon-based models are also subject to this approximation, the effect of a poor approximation of the ROC curve is unlikely to account for the observed magnitude of the difference in performance of the machine learning models and lexicon-based models.

With respect to the machine learning models, logistic regression achieved the highest median AUC score of 0.9010, followed closely by the ANN (with a median score of 0.9010), the CNN (with a median score of 0.8975), naïve Bayes (with a median score of 0.8930) and LSTM (with a median score of 0.8965), whilst the SVM achieved a slightly lower median AUC score of 0.7985. The variability of the scores is larger for the deep learning models than for the '*shallow*' learning models. This may be attributed to the fact that the variability of the latter models originates only from the varying data sets, whilst the deep learning models are also subject to variation in their initial weight parameters. Overall, reserving judgement on the SVM, the machine learning models achieved comparable performance on the data set in terms of the AUC score.

The performance achieved by the models in terms of accuracy is shown in Figure 7. As in the case of the AUC score, a pronounced distinction between the lexicon-based models and machine learning models is exhibited, with the former achieving median accuracy scores in the range of 39%–48% and the latter achieving median accuracy scores between 82% and 85%. In this case, however, the Sentiwordnet model appears to outperform the other three lexicon-based models by a considerable margin, followed by the Vader algorithm which, in turn, appears to outperform the remaining two lexicon-based models. It is interesting to note that these models achieved much higher accuracies in other problem domains. In the original paper in which Vader was presented, for instance, an accuracy of 96% was reported for social media texts, whilst 61% and 63% accuracies were reported for movie and product reviews, respectively [27].

Shifting the focus towards the machine learning models, the naïve Bayes algorithm appears to be slightly inferior to the other models, exhibiting the lower median accuracy of 82.20% and a large variance, with scores ranging between 79.60% and 83.60%. The SVM model, on the other hand, fares favourably in respect of this metric, achieving the highest median accuracy of 84.60%. The CNN, ANN and logistic regression models follow closely with median accuracies of 84.20%, 84.00% and 84.00%, respectively, whilst the LSTM network achieves a median accuracy of 83.30%. Overall, in terms of accuracy, it would appear that all machine learning models achieve comparable performance, with the SVM, the CNN, the ANN and logistic regression slightly outperforming naïve Bayes and LSTM.

Similar box plots were constructed to compare the performance of the various document representations. In Figure 8, the maximum[15] AUC scores achieved during each run are displayed in terms of both the document representation (with the exception of word em-

which first computes the probability of a point belonging to a certain class.

[15] As before, the maximum score achieved is selected for each experimental run, in this case across all algorithms, resulting in ten data points for each box plot.

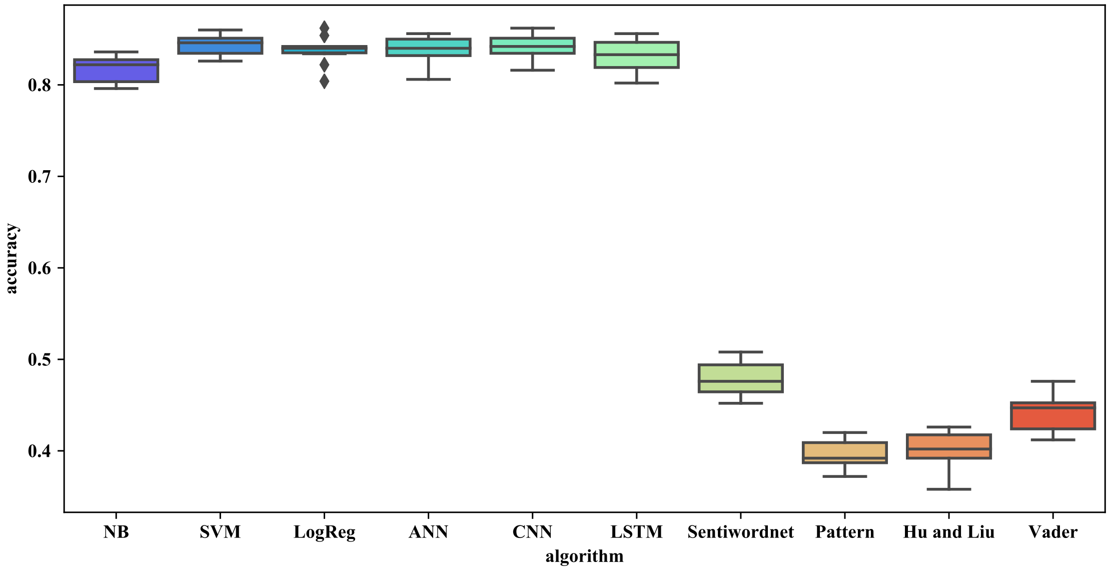


Figure 7: *Model performance in terms of accuracy. The accuracy values achieved during ten experimental runs are shown in the form of box plots.*

beddings) and the n-gram ranges of unigrams (denoted by $(1,1)$ in the figure), bigrams (denoted by $(2,2)$ in the figure) and unigrams with bigrams (denoted by $(1,2)$ in the figure).

From this figure it is clear that there is a significant decrease in performance when bigrams are used than when unigrams or a combination of unigrams and bigrams are used. The performance of the other two n-gram ranges, on the other hand, are competitive. Looking for common term collocations without regard for the individual words employed in a document therefore causes a decline in performance, whilst adding this information to the simple unigram representation does not seem to affect a significant increase in performance in this case. Furthermore, for both unigrams and unigrams with bigrams, the use of term presence and term frequency document models results in a similar performance, whilst the TF-IDF representation performs marginally worse. When only bigrams are employed, on the other hand, there is little distinguishable difference between the document models, save the fact that the TF-IDF representation appears to produce more stable AUC results across the experimental runs.

In respect of the accuracy score, the same large performance gap is exhibited between the bigrams-only n-gram range and the other two representations, as shown in Figure 9. For this metric, a more discernible difference was also found between the different document representations in each case. As shown in the figure, the TF-IDF representation achieved a higher median accuracy for both the unigram and the unigram with bigram n-gram ranges, whilst the term presence model appeared to perform slightly better for the bigram representation. Considering the importance of a word based on the frequency of its usage within a document and within the corpus as a whole could thus be beneficial for sentiment classification. It is likely that this pattern is not exhibited in the bigrams only case due to fewer appearances of each bigram in the corpus than individual unigrams.

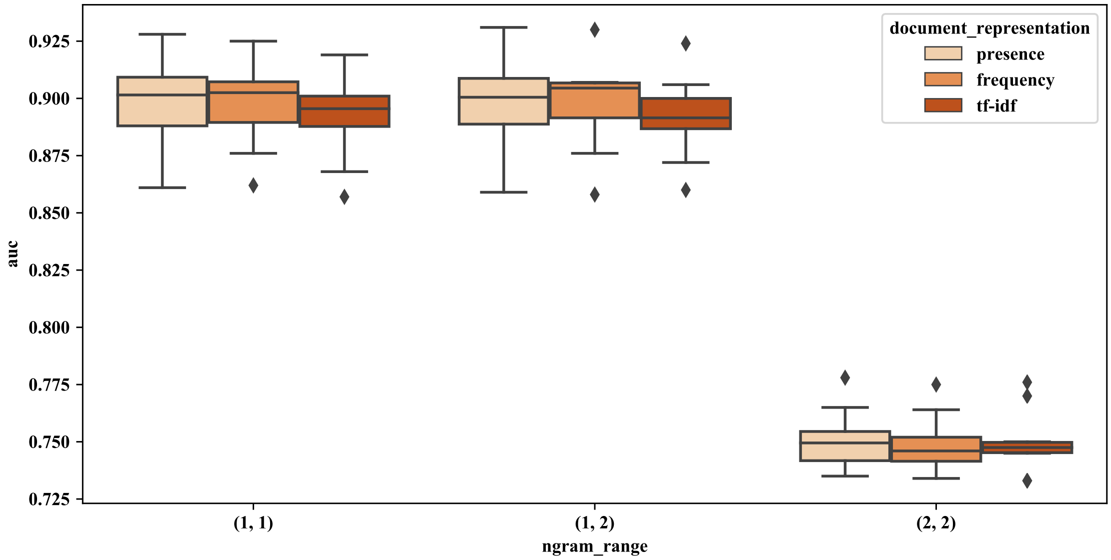


Figure 8: *The effect of feature engineering on the AUC value. The highest AUC value achieved by models using various document representations and n-gram ranges during ten experimental runs are shown in the form of a box plot in each case.*

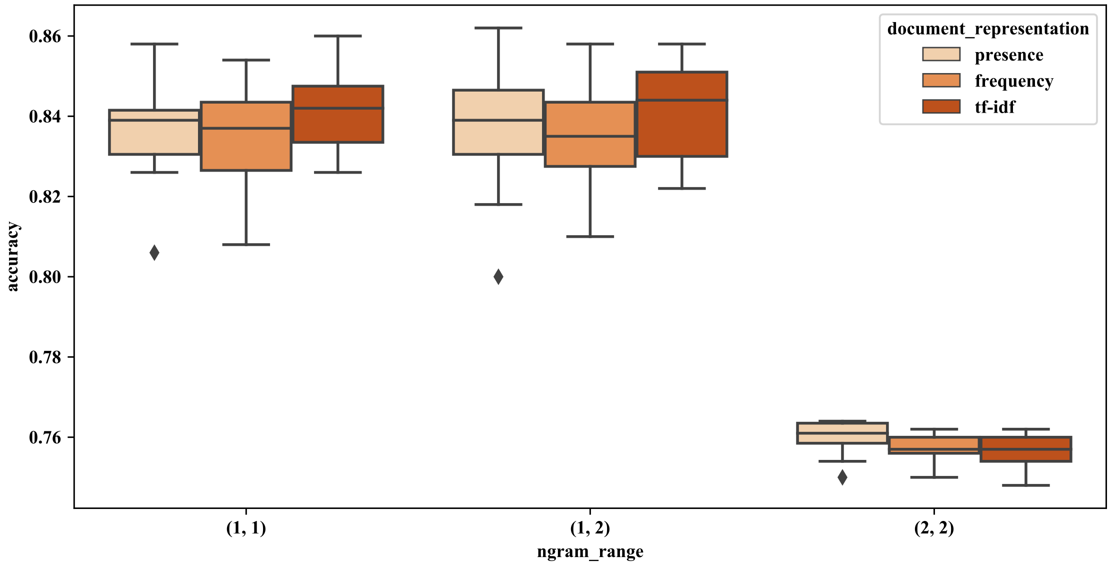


Figure 9: *The effect of feature engineering on accuracy. The highest accuracy values achieved by models using various document representations and n-gram ranges during ten experimental runs are shown in the form of box plots.*

In summary, the experiments showed that the off-the-shelf lexicon-based methods were significantly outperformed by the machine learning models developed in respect of the training data. The machine learning models achieved overall competitive performances, with the performance of the SVM and the naïve Bayes classifier proving slightly inferior in terms of the AUC score and accuracy, respectively. Based on both metrics, the top three performing algorithms were logistic regression, the CNN and the ANN in no discernible order. With respect to the input features, there is a clear decline in performance when bigrams are used in isolation without including unigrams in the feature vector. Differences in performance based on document models were more difficult to identify. It would appear that the use of the TF-IDF representation had a positive effect on the accuracy score, but a negative effect on the AUC score (particularly for unigrams and unigrams with bigrams). These differences were, however, marginal.

The classification results of the third-party software described in the previous section were also evaluated in respect of the labelled data. Unfortunately, the software's ratings were not available for all of the data. The evaluation was therefore carried out in respect of 2 486 (99.44%) of the 2 500 labelled reviews. The confusion matrix illustrating the results of the classification is shown in Figure 10.

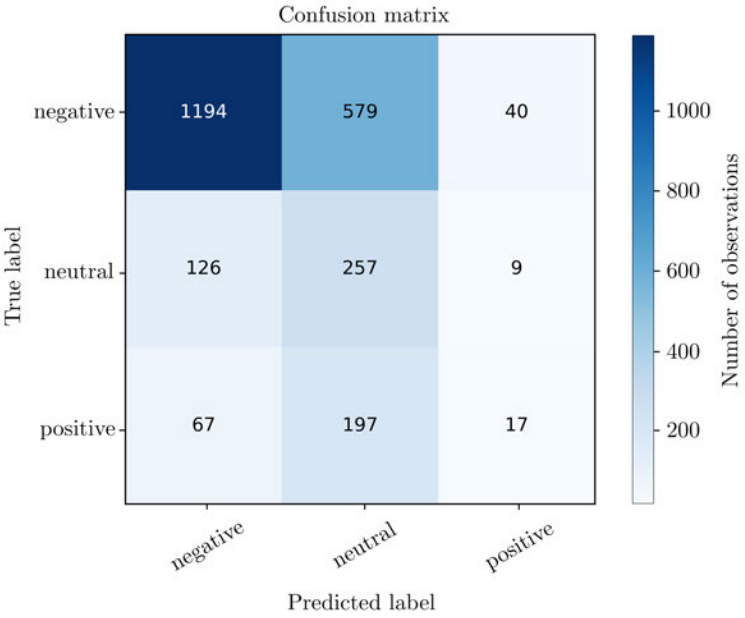


Figure 10: *The classification results of the third-party sentiment analysis software in the form of a confusion matrix.*

As is evident from the figure, the software often fails to distinguish between the negative and the neutral class, resulting in an accuracy score of 59.05% and a micro-weighted AUC score of 0.6602. Compared with the lexicon-based models evaluated during the case study, which achieved median AUC scores between 0.60 and 0.62 (with a maximum score of 0.649) and median accuracies between 39% and 48% (with a maximum score of 50.80%), the software thus fares favourably. In comparison with the custom machine learning models developed, however, the software does not achieve competitive results. The median scores for the AUC and accuracy score range between 0.79–0.91 and 82%–85% for these models, respectively. Since the software does not make use of any annotated training data from the industry partner, it likely makes use of either a lexicon-based model or a machine learning model that has been pre-trained in respect of other annotated data. The results achieved by this software in respect of the data from the industry partner further highlight the problem of applying a model developed within one specific context to data originating within another context. Off-the-shelf models or readily available software by a third-party vendor may produce adequate results if the application context is sufficiently similar to the development context. If this is not the case, however, a significant improvement in performance may be achievable by developing context-specific models according to the process outlined in this section.

4.3 Relating sentiment to review content and external attributes

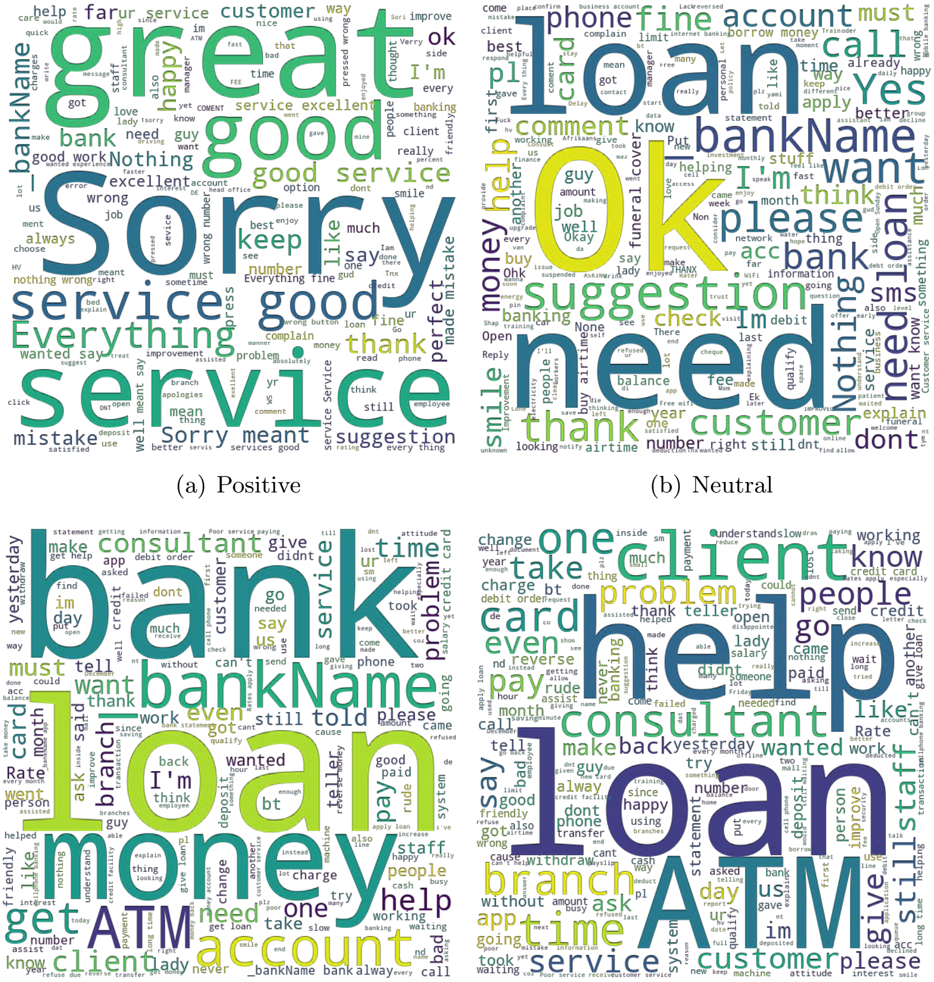
In §4.2, various models were developed and evaluated in respect of the labelled subset comprising 2 500 of the 10 636 available records. Based on its performance in respect of this subset, the CNN model was selected to classify the sentiment polarities of the remaining case study data. The hyperparameter values of this model are given in Table 2, where the number of training epochs were set to 12 by the 3-fold cross-validated grid search. According to this model, 75.10% of the total free-form text responses submitted by clients who rated their experience at the bank as a 2 or 3 have a negative sentiment polarity. Furthermore, 15.50% of responses bear no sentiment, whilst 9.39% have a positive sentiment polarity.

Whilst this sentiment distribution already gives some insight into the data (*e.g.* almost 10% of customers who submit a negative rating follow up with a positive comment), further analysis is required in order to gain actionable insight from the data. In this section, the relationship between the sentiment expressed in a document (as classified by the CNN model) and the content of the document is first explored by means of word cloud representations, topic modelling and the visualisation of word frequencies. Subsequently, the relationship between sentiment and the values of supplementary structured data, in this case the data describing the branch and customer associated with a review, is investigated by means of data visualisation.

4.3.1 Analysing the content of reviews

An illustration of the word usage per sentiment class is given in the form of word cloud representations in Figure 11. Examining the word cloud for the *positive* sentiment class in Figure 11(a) reveals common phrases such as "*sorry [I] meant*," "*made mistake*" and "*wanted [to] say*," which underscore the notion that the rating scale was misinterpreted by these customers. Other common sentiments are reflected in the phrases "*everything fine*," "*nothing wrong*," and "*good service*." Customers in this review category therefore seem to be satisfied overall with the bank and its products and services.

The word cloud of the *neutral* responses in Figure 11(b) is more difficult to interpret. The most common terms appear to be *loan*, *Ok* and *need*. The phrase "*need loan*" is also a frequent collocation. Responses in this category thus often seem to refer to loan requests. Other phrases, such as "*smile*," "*funeral cover*" and "*buy airtime*," however, suggest a wide variety of topics.



(a) Positive

(b) Neutral

(c) Negative

(d) Negative, with certain terms excluded

Figure 11: *Word cloud representations of the reviews in each sentiment class.*

The following five random samples of reviews in this sentiment class were drawn from the corpus in order to gain a better understanding of the subject matter:

(i) "*You no better I dont*",

(ii) "*Ok*",

(iii) "*Greetings and smile please*",

(iv) "*Triple repayment*", and

(v) "*Open the business accounts*".

Once again, a variety of topics is discussed without a clear sentiment orientation and with little actionable information. Those comments classified as *neutral* by the model therefore appear to constitute short answers that bear little information or sentiment and are often difficult to contextualise.

Finally, the word cloud illustrating the word usage amongst *negative* reviews in Figure 11(c) is dominated by generic terms, such as *bank*, *money*, *account* and the name of the bank (*_bankName*). In order to better analyse the meaningful words in this context, these words were excluded from the word cloud along with other non-informative words, such as *get*, *want* and *I'm*, resulting in the word cloud shown in Figure 11(d). From this figure it is now evident that the most important keywords mentioned in negative reviews are *loan* and *ATM*, and that many customers also made reference to *help* in their negative response. Other frequent terms that may bear some information include *time*, *app*, *service*, *staff* and *card*.

This elementary analysis sheds some light onto the contents of customer complaints. A topic analysis was also performed in order to gain a deeper insight into these contents before the customers' most important points of concern were further explored. To this end a *Latent Dirichlet Allocation* (LDA) topic model (see [7]) was fit to the data in order to identify groups of frequently co-occurring words in the corpus. The bag of words representation of the review documents formed the input to this model, where only terms occurring in at least two documents and no more than 50% of the documents were retained in an attempt to filter out common, uninformative words. The resulting model was then visualised by means of the visualisation package *LDAvis* by Sievert and Shirley [46].

The resulting visualisation for the data with five topics and two iterations is shown in Figure 12. In the left-hand plot, each of the topics is represented by a circle in a two-dimensional plot. The centres of the circles are determined by computing the distance[16] between the two topics according to the term-topic distributions learnt during the training of the LDA model. This distance is scaled to two dimensions by means of PCA. The area of each of the circles is proportional to $N_k/\sum_k N_k$, where N_k is the estimated number of tokens that were generated by topic k across the entire corpus.

The terms on the right-hand side of Figure 12 represent the thirty most *salient* terms in the corpus, where saliency is calculated according to the formula developed by Chuang *et al.* [11] as

$$s(w) = p_w \sum_{K} P(k \mid w) \log\left(\frac{P(k \mid w)}{p_w}\right),$$

with p_w and $P(k \mid w)$ representing the probability of observing a word w in the entire corpus and in the subset of documents belonging to topic k, respectively. The width of the bar next to each term is scaled according to the estimated total number of occurrences of term w in the corpus. The *relevance* of a term w to topic k may, furthermore, be calculated as [46]

$$r(w, k \mid \lambda) = \lambda \log(\phi_{kw}) + (1-\lambda) \log\left(\frac{\phi_{kw}}{p_w}\right),$$

where λ is a user-specified parameter and ϕ_{kw} is the estimated probability that term w is generated by topic k. Upon exploring the topics and important terms in the corpus, insight can be gained into the frequency with which these topics and terms were present in the data.

[16] In this case, Jensen-Shannon divergence is adopted as a measure of the similarity between the term-topic probability distributions. Details of this method may be found in [20].

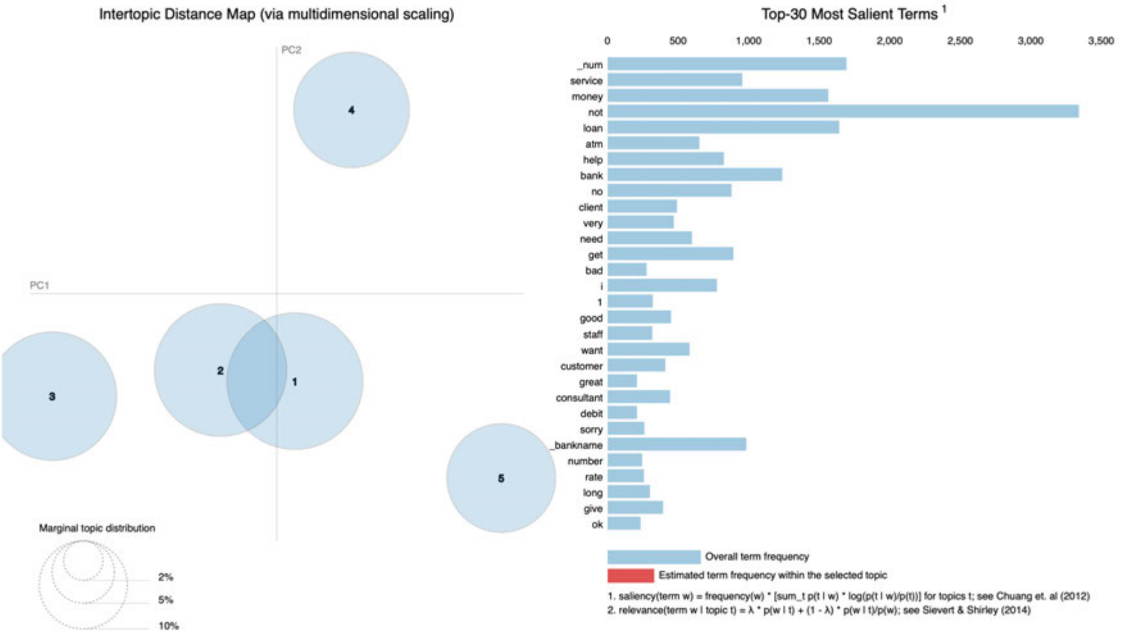


Figure 12: *The results of the LDA topic model based on five topics and two passes through the case study data.*

From the figure, it may be deduced that the five discovered topics are relatively well separated with the exception of Topics 1 and 2, which overlap considerably. Topics 1–4 and 5 are distributed along the first principal component (the horizontal axis), while exhibiting similar values for the second principal component (the vertical axis). Topic 4, however, is clearly distinguished from the remaining topics by its value for the second principal component. Furthermore, Topics 1–3 appear in the corpus with relatively equal frequencies (in 20%–22% of the documents), whilst Topics 4 and 5 appear less frequently (in approximately 17% and 15% of the documents, respectively).

Based on both the frequency with which each of the most salient terms in the corpus were observed in a given topic and the relevance of words to a given topic, several important keywords could be associated with each topic. These associations are given in Table 3. It was furthermore deduced, based on the results in the table and the relative frequencies of these terms in each of the topics, that Topic 1 is concerned with general inquiries related to money, rates and accounts at the bank. Topic 2, on the other hand, is primarily related to ATMs. Topic 3 comprises matters pertaining to loans, staff and consultants, along with which reference is typically also made to the customer or client. Due to its association with primarily numerical characters, the rating indices 1 and 2 that were excluded from the numerical grouping step during preprocessing and the terms *rate* and *sorry*, Topic 4 was deemed to relate to cases in which customers misunderstood the rating scale and were, in fact, correcting this misunderstanding. Finally, Topic 5 is related to the bank's service and debit orders, which appear to be described frequently by adjectives such as *good*, *great*, and *bad*, and the adverb *very*.

Topic	Frequent *salient* keywords	*Relevant* keywords
1	money, get, good, rate, not	no, money, account, bank, get, people, good
2	ATM, money, help, no, want, long, ok	no, not, ATM, bank, help, _bankname, want, money
3	loan, help, client, get, staff, customer, consultant, give	loan, not, get, need, client, consultant, customer, staff
4	_num, 1, sorry, number, rate, long	_num, loan, card, 1, 2, number, sorry
5	service, very, bad, good, great, debit	service, bank, very, _bankname, bad, good, great, debit

Table 3: *The keywords associated with each topic in Figure 12 based on both the frequency with which salient words were observed in and the relevance of words to a given topic.*

In order to determine which of these concerns are the most pressing for the customers, the number of reviews in each sentiment class that contain the keywords or noun phrases identified from the LDA topic model and the word cloud in Figure 11(d) were compared by means of a count plot. The resulting plot is shown in Figure 13. From this graph, it is clear that *money, loan, help, ATM* and *service* are the most prominent keywords from the five extracted topics, whilst the keyword *time* (which does not feature in the topic analysis) is also relatively prominent. It is interesting to note here that *service* is the only keyword mentioned in almost as many positive reviews as negative reviews.

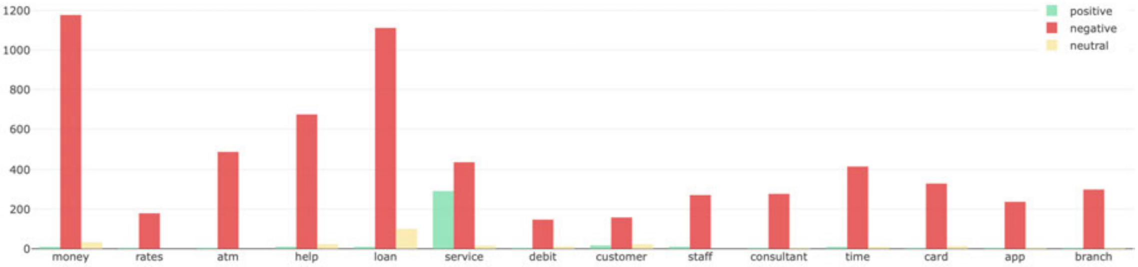


Figure 13: *The frequency with which selected keywords were observed in the corpus.*

The documents in the corpus were subsequently filtered according to the words they contain in order to gain further insight into what, in particular, customers may have been dissatisfied with in relation to the above-mentioned *prominent* keywords (those that were observed in more than 400 negative reviews), with the addition of the keywords *staff* and *consultant*, which relate to the same concept and have a combined observed frequency greater than 400. To this end, the word clouds of the negative reviews in each filtered selection were scrutinised for informative words and phrases related to the keyword. Furthermore, a few random samples of original reviews were drawn from each selection in order to contextualise these phrases.

Among the most frequently used phrases in reviews containing the keyword *loan*, for example, were *get*, *want*, *need*, *help* and *give*, as well as *qualify*, *declined*, *refused* and *apply*. This suggests that many customers are concerned that they did not qualify for or receive a loan. It was, furthermore, found that the keyword *money* was used in a variety of contexts, but that customers often mentioned having trouble *reversing* money, *drawing* money at ATMs or *borrowing* money in the form of personal loans. Many customers also complained about *needing* money, although this provides little actionable insight for the bank.

The word *help* typically appeared in negative reviews to refer to a lack of assistance received from staff members, especially in acquiring a loan, and to complain about unfriendly service. The related keyword *staff* was similarly associated with complaints of unfriendly service from poorly trained employees, as well as too few staff members. These sentiments were shared in reviews mentioning the keyword *consultant*, in which staff members were described as unprofessional, uninformed and rude with several complaints mentioning that consultants were preoccupied with personal conversations amongst themselves. Finally, reviews which contain the word *service* reflected similar issues. Interestingly, many of the sampled negative reviews in this selection described having had both positive and negative experiences related to customer service. This impression is strengthened by the fact that, although only ratings of 2 and 3 were followed up on during the customer survey, almost half of the reviews mentioning service were, in fact, positive in sentiment, describing the service as *great* or *excellent.* It may thus be concluded that customer service in the bank is inconsistent over branches.

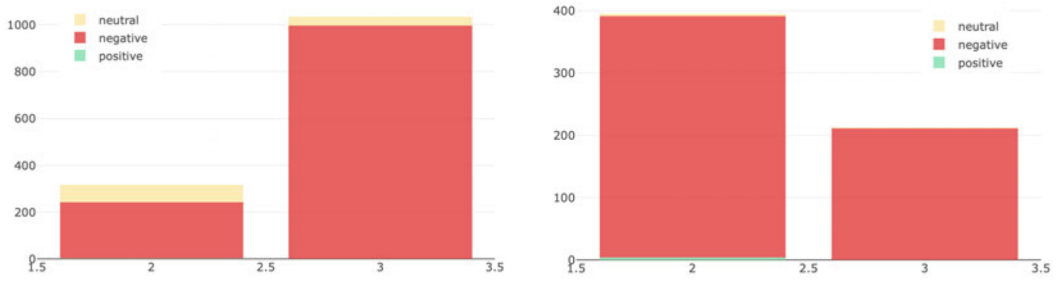
With regards to *ATM*s, customers were found to complain primarily about the insufficient number of available ATMs, especially those with *depositing* capabilities, as well as about the associated long queues and waiting times. Furthermore, customers lamented that machines were frequently *slow*, *faulty* or *offline.* Finally, several customers requested additional security measures in the vicinity of ATMs due to frequent cases of robbery.

The final *prominent* keyword investigated was *time.* It became clear during the analysis, however, that the high frequency of this term was not due to its use as some indicator of the bank's performance but rather as a generic term in several different contexts (*e.g. "every time," "place and time," "the machine timed out,"* and "*the staff took their time.*") This keyword was therefore not further investigated.

4.3.2 Analysing the relationship between sentiment and structured attributes

Having successfully established the contents of the customer complaints, the relationship between the distribution of sentiment polarity and the structured supplementary data was investigated next. Due to the strong bias towards the negative sentiment class present in the data, this analysis proved challenging. In fact, for most of the examined variables, the distribution of sentiment polarity was approximately equal across all possible categories. By filtering the corpus according to the keywords mentioned in reviews, however, some insight could still be gained as to the dominant characteristics of the customers who submitted negative reviews and the branches that they visited prior to submitting these reviews.

The numerical ratings submitted by customers prior to the unstructured review (the *Q01 Values*) are, for instance, distributed differently with respect to customers who mentioned different keywords in their reviews. From Figure 14(a), for example, it is clear that most customers who complained about loans rated the bank with a 3 (the worst possible rating). This keyword, being among the three most frequently mentioned keywords, therefore also appears to be a considerable source of dissatisfaction for customers. The keyword *ATM*, on the other hand, while frequently mentioned by customers, did not appear to result in as negative an overall rating. In fact, as shown in Figure 14(b), most customers who mentioned ATMs in their subsequent reviews evaluated the bank with a rating of 2 out of 3. By analysing this attribute for subsets of customers who mentioned different keywords, it could be gauged how important certain topics are for overall customer satisfaction.



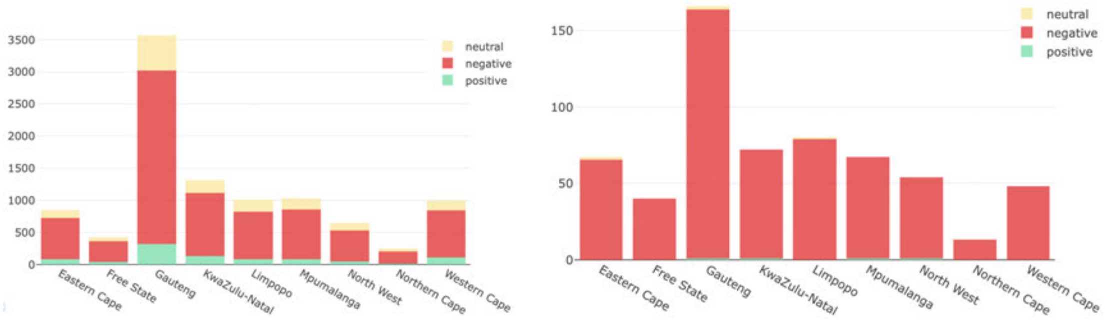
(a) loan (b) ATM

Figure 14: *Sentiment counts by Q01 Value for reviews that contain the keywords (a) loan and (b) ATM.*

A bubble map representation of the case study data was generated in an attempt to identify trends between sentiment and location. This representation, however, showed a clear dominance of negative sentiment throughout the country with little evidence to suggest that customers in any particular area were more likely to submit positive reviews following their non-positive rating. In fact, the distribution of reviews was consistent with the population density of South Africa. Since a greater number of reviews can be expected from more densely populated areas, no unusual hotspots of dissatisfied customers could be identified by means of this graph.

At a higher level of abstraction, count plots visualising the number of reviews in each sentiment category by province were generated, as shown in Figure 15(a). From this representation, it is clear that the majority of reviews were associated with branches located in Gauteng, followed by KwaZulu-Natal with less than half the number of associated reviews. The smallest numbers of reviews are associated with the Free State and the Northern Cape.

Since not all branch and customer records were available for this case study, it is unclear how this distribution compares with the overall distribution of the industry partner's branch or customer locations. It is, however, interesting that a similar distribution was uncovered for reviews containing the keywords *loan, service, consultant* and *help*, whilst the numbers of reviews containing the keyword *ATM* appear to be distributed differently across provinces. From Figure 15(b), it is clear that the Eastern Cape, Free State, KwaZulu-Natal, Limpopo, Mpumalanga and North-West provinces are all more strongly



(a) All reviews

(b) Keyword ATM

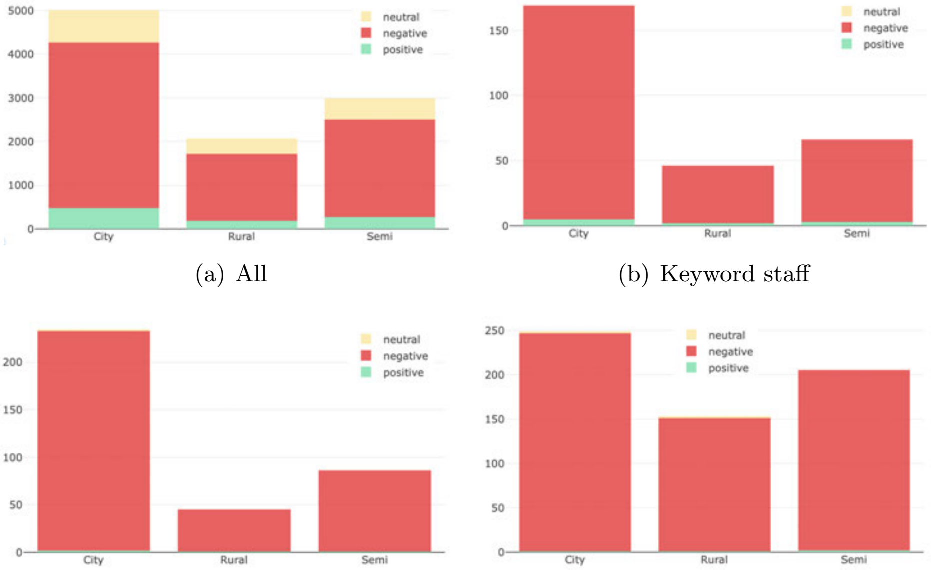
Figure 15: *Sentiment counts by branch province for (a) all reviews and (b) reviews that contain the keyword ATM .*

represented by reviews related to ATMs than in the general case in Figure 15(a). It is possible that the bank has expended fewer resources to install and service ATMs in less densely populated areas, resulting in a higher relative level of customer dissatisfaction with respect to this topic in the affected provinces.

On a similar note, the distribution of reviews with respect to branch types was investigated. Of all reviews analysed, approximately 50% were associated with urban branches, 20% were associated with branches in rural areas and 30% were associated with branches in semi-rural areas, as illustrated in Figure 16(a). A similar distribution was exhibited for reviews mentioning the keywords *loan*, *service* and *help*. As shown in Figures 16(b) and 16(c), however, the proportion of reviews associated with urban branches rose to over 60% and 64% for reviews pertaining to *staff* and *consultants*, respectively. The behaviour of customer service staff thus seems to be a greater cause for concern in metropolitan areas. More interestingly, the proportions of reviews associated with urban, rural and semi-rural branches are 40%, 25% and 35%, respectively, for reviews that mention ATMs. This confirms the theory that ATM machines pose a significant problem in rural and semi-rural areas.

The attributes describing the customers themselves include demographic information such as their town and country of residence, as well as information related to the banking profile, including the customers' service plan with the industry partner, average monthly bank fee, and loan status. Many of these attributes did not reveal particularly interesting insights, since, as previously mentioned, the attributes of the bank's average customer were not available for comparison. Some attributes, however, revealed interesting patterns when compared to the subsets of reviews that mention certain keywords.

The median of the average monthly bank fee paid by customers who mentioned ATM in their reviews, for example, was R87, compared with the R70–R75 median fees paid monthly by customers complaining of other matters. These customers, who complained primarily of a lack of ATMs, thus incurred an increase in fees of over 15% — likely because of surcharges incurred for drawing money at the machines of other banks.



(a) All

(b) Keyword staff

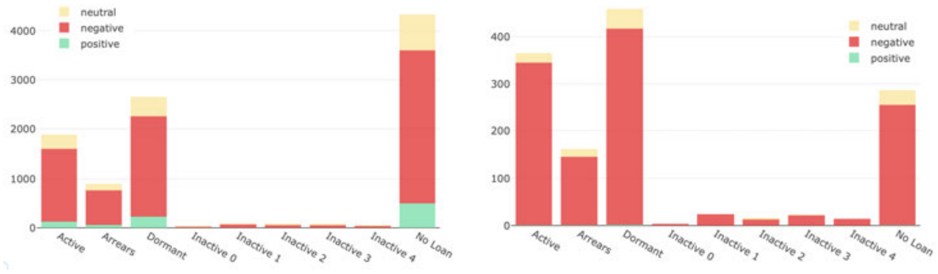
(c) Keyword consultant

(d) Keyword ATM

Figure 16: *Sentiment counts by branch type for (a) all reviews, (b) reviews that contain the keyword staff, (c) reviews that contain the keyword consultant and (d) reviews that contain the keyword ATM.*

Furthermore, the median salary of all surveyed customers was R6 475 and varied between R6 368 and R6 774 for the keywords *ATM*, *loan*, *help* and *service*. The median salary earned by customers complaining about *staff* and *consultants*, on the other hand, was R7 909 and R9 000, respectively. Higher earning customers therefore seemed to be more concerned with the attitude and aptitude of customer contact employees than customers who earn less.

Finally, the attribute describing a customer's loan status revealed a similar pattern to that of the general case shown in Figure 17(a) for most keywords. As is evident from the figure, most clients (43%) who reviewed the bank did not have a loan at the time of the survey. In contrast, 19% of customers did have active loans, whilst 9% were in arrears with their loan repayments and a further 9% and 3% of customers had loans that were dormant or in some form inactive, respectively. Focusing exclusively on customers who submitted reviews that mention the keyword *loan* produced a different distribution. As shown in Figure 17(b), the majority of these customers (33%) had loans that were classified as dormant, whilst a further 26% had active loans, 6% had inactive loans and 12% were falling short on loan repayments. The proportion of customers without loans was 22% in this case. Most of the customers who complained about loans thus appear to be those customers who had, in some form or another, not been able to keep up to date with their existing loan repayments. This information, in conjunction with the content analysis performed previously in respect of this subset of reviews, leads to the conclusion that many customers are dissatisfied that they could not acquire a *second* loan from the bank.



(a) All

(b) Keyword loan

Figure 17: *Sentiment counts by loan status for (a) all reviews and (b) reviews that contain the keyword loan.*

5 Conclusion

The aim in this paper was to demonstrate in the form of a case study how a raw data set of unstructured customer reviews may be evaluated in the South African banking sector. The preprocessing performed on the data were first described, illustrating the extent to which each of the filtering and normalisation steps contributed to the reduction of the vocabulary size, resulting in a final vocabulary that was more than 60% smaller than the original, unprocessed vocabulary. Subsequently, the model development process was delineated, including the approach taken to tune various hyperparameters and the evaluation of machine learning models trained in respect of the case study data, as well as off-the-shelf lexicon-based models and the model employed by the third-party vendor contracted by the industry partner. Finally, the results of the CNN model were analysed, revealing the contents of the customer reviews, providing insight into the importance of various topics to customers and attempting to find associated patterns in branch and customer attributes.

The reviews exhibited poor grammatical structure, frequent misspellings and colloquialisms, as well as a mixture of various languages. Furthermore, due to the process by which the data were collected, a strong bias towards the negative sentiment class was present in the data. In spite of these challenges, the newly generated machine learning models achieved median AUC scores of up to 0.9 over ten replications. Furthermore, these models significantly outperformed commercial tools and existing lexicon-based models from the literature in terms of both the AUC score and the classification accuracy. This may be attributed to the fact that these models were developed within a different context that does not correlate well with the nuances of the South African and/or banking domain. The computational learning approach, in contrast, appeared to successfully identify context-specific patterns in the data.

Furthermore, it was demonstrated that a combination of topic modelling and visualisation techniques can be applied to the results of a sentiment classifier in order to effectively extract insights from the data. By analysing the contents of customer reviews in combination with the sentiment classes, it was determined that almost 10% of customers who had rated the bank negatively actually followed up with positive reviews, often due to having misunderstood the rating scale. Furthermore, the most prominent points of concern for customers were identified as matters related to a lack of assistance in acquiring personal loans, insufficient or faulty ATMs, and poorly trained or unprofessional staff. Finally, by analysing the available structured variables in conjunction with these issues, it was found that complaints about a lack of ATMs appear to be of greater concern in rural and semi-rural areas, and that these appear to come from customers incurring, on average, over 15% higher monthly fees than other customers. Furthermore, loans were a subject of complaint primarily for customers with existing loans, whilst staff behaviour appeared to be an issue for higher-earning customers from urban areas. These insights may be used to drive decision-making in the bank. Targeted resources may, for example, be deployed to address cash drawing facilities for customers in rural areas, improve the support structure for existing loan recipients and retrain employees servicing customers in more affluent areas.

6 Future work

In this case study, only pre-trained lexicon-based models for sentiment analysis were considered. Most of these models make use of sentiment lexicons that were generated using general-purpose English dictionaries or lexical resources, such as WordNet. One of these models, Vader, also included common expressions used in the context of social media. None of these sentiment lexicons were generated in the context of the South African banking sector, however. This poses the question whether a lexicon-based model making use of a sentiment lexicon generated for the specific context would achieve better performance in respect of the case study data than pre-trained models which faired poorly in comparison with machine learning models.

It would also be interesting to repeat the case study analysis in another context in order to determine whether machine learning models still significantly outperform lexicon-based models when applied to data from a different domain. Since the Vader algorithm was developed in the context of micro-blogging, a comparison in this domain would be of particular interest. Comparing various approaches (*i.e.* lexicon-based approaches, '*shallow*' machine learning approaches and deep learning approaches) directly in respect of several benchmark data sets may, furthermore, allow for general recommendations to be made in respect of choosing an appropriate modelling approach for a particular problem setting.

References

[1] Annett M & Kondrak G, 2008, *A comparison of sentiment analysis techniques: Polarizing movie blogs*, Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence, Windsor, pp. 25–35.

[2] Aronof M & Fudeman K, 2011, *What is morphology?*, Wiley-Blackwell, Hoboken (NJ).

[3] Atkinson K, 2016, *GNU Aspell* , [Online], [Cited May 2019], Available from http://aspell.net/.

[4] Bespalov D, Bai B, Shokoufandeh A & Qi Y, 2011, *Sentiment classification based on supervised latent n-gram analysis*, Proceedings of the 20th ACM International Conference on Information and Knowledge Management, Glasgow, pp. 375–382.

[5] Bhuta S & Doshi U, 2014, *A review of techniques for sentiment analysis of Twitter data*, Proceedings of the International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, pp. 583–591.

[6] Blake C, 2011, *Text mining*, Annual Review of Information Science and Technology, **45(1)**, pp. 121–155.

[7] Blei DM, Ng AY & Jordan MI, 2003, *Latent Dirichlet allocation*, Journal of Machine Learning Research, **3**, pp. 993–1022.

[8] Cambria E, Schuller B, Xia Y & Havasi C, 2013, *New avenues in opinion mining and sentiment analysis*, IEEE Intelligent Systems, **28(2)**, pp. 15–21.

[9] Chaovalit P & Zhou L, 2005, *Movie review mining: A comparison between supervised and unsupervised classification approaches*, Proceedings of the 38th Annual Hawaii International Conference on System Sciences, Big Island (HI), pp. 1–9.

[10] Choi Y, Kim Y & Myaeng S-H, 2009, *Domain-specific sentiment analysis using contextual feature generation*, Proceedings of the 1st International CIKM Workshop on Topic-sentiment Analysis for Mass Opinion, Hong Kong, pp. 37–44.

[11] Chuang J, Manning CD & Heer J, 2012, *Termite: Visualization techniques for assessing textual topic models*, Proceedings of the International Working Conference on Advanced Visual Interfaces, Capri, pp. 74–77.

[12] Cieliebak M, 2018, *Sentiment analysis: Distinguish positive and negative documents*, [Online], [Cited July 2019], Available from `http://www:agilemodeling:com/style/classDiagram.htm`.

[13] Claesen M & de Moor B, 2015, *Hyperparameter search in machine learning*, Proceedings of the 11th Metaheuristics International Conference, Agadir, pp. 1–5.

[14] Computational Linguistics & Psycholinguistics Research Center, 2018, *Pattern*, [Online], [Cited July 2019], Available from `https://www:clips.uantwerpen.be/pattern`.

[15] Dave K, Lawrence S & Pennock DM, 2003, *Mining the peanut gallery: Opinion extraction and semantic classification of product reviews*, Proceedings of the 12th International World Wide Web Conference, Budapest, pp. 519–528.

[16] Devika MD, Sunitha C & Ganesh A, 2016, *Sentiment analysis: A comparative study on different approaches*, Procedia Computer Science, **87**, pp. 44–49.

[17] Dhaoui C, Webster CM & Tan LP, 2017, *Social media sentiment analysis: Lexicon versus machine learning*, Journal of Consumer Marketing, **34(6)**, pp. 480–488.

[18] Esuli A & Sebastiani F, 2006, *SentiWordNet: A publicly available lexical resource for opinion mining*, Proceedings of the 5th Conference on Language Resources and Evaluation, Genoa, pp. 417–422.

[19] Fawcett T, 2006, *An introduction to ROC analysis*, Pattern Recognition Letters, **27(8)**, pp. 861–874.

[20] Fuglede B & Topsoe F, 2004, *Jensen-Shannon divergence and Hilbert space embedding*, Proceedings of the International Symposium on Information Theory (ISIT), Chicago (IL), p. 31.

[21] Gamon M, Aue A, Corston-Oliver S & Ringger E, 2005, *Pulse: Mining customer opinions from free text*, Proceedings of the 6th International Symposium on Intelligent Data Analysis, Madrid, pp. 121–132.

[22] Goodfellow I, Bengio Y & Courville A, 2016, *Deep learning*, MIT Press, Cambridge (MA).

[23] Google, 2013, *word2vec* [Online Code archive], [Cited August 2018], Available from https://code.google.com/archive/p/word2vec/.

[24] Hailong Z, Wenyan G & Bo J, 2014, *Machine learning and lexicon based methods for sentiment classification: A survey*, Proceedings of the 11th Web Information System and Application Conference, Tianjin, pp. 262–265.

[25] Hoopr R & Paice C, 2005, *The Lancaster stemming algorithm*, [Online], [Cited December 2019], Available from http://www.comp.lancs.ac.uk/computing/research/stemming/index.htm.

[26] Hu X & Liu H, 2012, *Text analytics in social media*, pp. 385–414 in Aggarwal CC & Zhai C (Eds), *Text mining*, Springer Science & Business Media, New York (NY).

[27] Hutto CJ & Gilbert E, 2014, *Vader: A parsimonious rule-based model for sentiment analysis of social media text*, Proceedings of the 8th International AAAI Conference on Weblogs and Social Media, Ann Arbor (MI), pp. 216–225.

[28] Kenyon-Dean K, Ahmed E, Fujimoto S, Georges-Filteau J, Glasz C, Kaur B, Lalande A, Bhanderi S, Belfer R & Kanagasabai N, 2018, *Sentiment analysis: It's complicated!*, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans (LA), pp. 1886–1895.

[29] Kingma DP & Ba J, 2015, *Adam: A method for stochastic optimization*, Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego (CA).

[30] Kiritchenko S, Zhu X & Mohammad SM, 2014, *Sentiment analysis of short informal texts*, Journal of Artificial Intelligence Research, **50**, pp. 723–762.

[31] le Cun Y, Bengio Y & Hinton G, 2015, *Deep learning*, Nature, **521**, pp. 436–444.

[32] Li F, Huang M & Zhu X, 2010, *Sentiment analysis with global topics and local dependency*, Proceedings of the AAAI Conference on Artificial Intelligence, Altanta (GA), pp. 1371–1376.

[33] Liu B, 2012, *Sentiment analysis and opinion mining*, Synthesis Lectures on Human Language Technologies, **1(5)**, pp. 1–168.

[34] Liu B & Zhang L, 2012, *A survey of opinion mining and sentiment analysis*, pp. 415–463 in Aggarwal CC & Zhai CX (Eds), *Mining text data*, Springer Science & Business Media, Berlin.

[35] Manning, CD, Raghavan P & Schütze, H, 2008, *Introduction to information retrieval*, Cambridge University Press, New York (NY).

[36] Medhat W, Hassan A & Korashy H, 2014, *Sentiment analysis algorithms and applications: A survey*, Ain Shams Engineering Journal, **5(4)**, pp. 1093–1113.

[37] Miller GA, 1995, *WordNet: A lexical database for English*, Communications of the ACM, **38(11)**, pp. 39–41.

[38] Moraes R, Valiati JF & Gavião Neto WP, 2013, *Document-level sentiment classification: An empirical comparison between SVM and ANN*, Expert Systems with Applications, **40(2)**, pp. 621–633.

[39] Mozeti I, Grčar M & Smailović J, 2016, *Multilingual Twitter sentiment classification: The role of human annotators*, PLoS ONE, **11(5)**, pp. 1–26.

[40] Pang B & Lee L, 2008, *Opinion mining and sentiment analysis*, Foundations and Trends in Information Retrieval, **2(2)**, pp. 1–135.

[41] Pang B, Lee L & Vaithyanathan S, 2002, *Thumbs up? Sentiment classification using machine learning techniques*, Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP), **10(7)**, pp. 79–86.

[42] Poria S, Cambria E, Winterstein G & Huang GB, 2014, *Sentic patterns: Dependency-based rules for concept-level sentiment analysis*, Knowledge-Based Systems, **69(1)**, pp. 45–63.

[43] Portilla JM, 2017, *Python for data science and machine learning bootcamp*, [Online Course], Udemy.com, Available from http://udemy.com/python-for-data-science-and-machine-learning-bootcamp.

[44] Ravi K & Ravi V, 2015, *A survey on opinion mining and sentiment analysis: Tasks, approaches and applications*, Knowledge-Based Systems, **89**, pp. 14–46.

[45] Sharma A & Dey S, 2012, *A comparative study of feature selection and machine learning techniques for sentiment analysis*, Proceedings of the 2012 ACM Research in Applied Computation Symposium (RACS), San Antonio (TX), pp. 1–7.

[46] Sievert C & Shirley K, 2014, *LDAvis: A method for visualizing and interpreting topics*, Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces, Baltimore (MD), pp. 63–70.

[47] Snowball, 2003, *Stemming algorithms for use in information retrieval*, [Online], [Cited December 2019], Available at http://www.snowball.tartarus.org/.

[48] Sperling G, 2017, *South African report delivers insight into influence of mobile on consumer behaviour* , [Online], [Cited February 2018], Available from http://www.biznisafrica.com/-south-african-reportdelivers-insight-influence-mobileconsumer-behaviour/.

[49] Van Rijsbergen CJ, Robertson SE & Porter MF, 1980, *New models in probabilistic information retrieval*, British Library Research and Development Department, London.

[50] TensorFlow, 2018, *Vector representations of words*, [Online], [Cited August 2018], Available from https://www.tensorflow.org/tutorials/representation/word2vec.

[51] TensorFlow, 2019, *TensorBoard: Visualizing learning*, [Online], [Cited July 2019], Available from https://www.tensorflow.org/guide/summaries%5C%20and%5Ctensorboard.

[52] Tripathy A, Agrawal A & Rath SK, 2016, *Classification of sentiment reviews using n-gram machine learning approach*, Expert Systems with Applications, **57**, pp. 117–126.

[53] Tsytsarau M & Palpanas T, 2012, *Survey on mining subjective data on the web*, Data Mining and Knowledge Discovery, **24(3)**, pp. 478–514.

[54] Wang S & Manning C, 2012, *Baselines and bigrams: Simple, good sentiment and topic classification*, Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Jeju Island, pp. 90–94.

[55] Weiss SM, Indurkhya N & Zhang T, 2010, *From textual information to numerical vectors*, pp. 13–38 in Indurkhya N, Weiss SM & Zhang T (Eds), *Fundamentals of predictive text mining*, Springer, London.

[56] Weiss SM, Indurkhya N, Zhang T & Damerau FJ, 2005, *Overview of text mining*, pp. 1–13 in Zhang T, Damerau F, Indurkhya N & Weiss SM (Eds), *Text mining: Predictive methods for analyzing unstructured information*, Springer, London.

[57] Yadollahi A, Shahraki AG & Zaiane OR, 2017, *Current state of text sentiment analysis from opinion to emotion mining*, ACM Computing Surveys, **50(2)**, pp. 1–33.

[58] Zhang L,Wang S & Liu B, 2018, *Deep learning for sentiment analysis: A survey*, Data Mining and Knowledge Discovery, **8(4)**, pp. e1253:1–e1253:34.