Sentiment Classification and Prediction of Job Interview Performance

Sarah S. Alduayj
School of Computer Science
University of Birmingham
Birmingham, United Kingdom
Email: sxa1115@student.bham.ac.uk

Phillip Smith
School of Computer Science
University of Birmingham
Birmingham, United Kingdom
Email: p.smith.7@cs.bham.ac.uk

Abstract— **Attracting and hiring talented employees is a challenge for companies. The job interview process is a very critical step for both employer and candidate. Having a smooth hiring process in a company will increase future employees' satisfaction. Candidates tend to share their feedback and experience of interviews and company's hiring process with others. Having a negative experience can affect its brand image and reputation as an employer. This will make it hard to attract talented employees. In this research, machine learning and neural network models, such as support vector machines, logistic regression, Naïve Bayes, and long short–term memory (LSTM), were trained to predict the candidates' sentiments after a job interview. Each model was trained using several data representations and weighting approaches, such as term binary, term frequency, and term frequency–inverse document frequency (TF–IDF). As a result, training logistic regression with TF–IDF and unigram word representation achieved an F1-measure of 0.814.**

Keywords- Logistic regression; Support vector machine; Random forest; K nearest neighbour; Naïve Bayes; Long short term memory; Skip gram; Continues bag of words;

I. Introduction

The employee life cycle within a company can be divided into six stages (attraction, recruitment, onboarding, development, retention and separation) [1][2]. Companies understand the effect of each stage on employees' engagement and devotion, where low employee engagement will have a negative effect on company brand and reputation. Human resources (HR) professionals will usually collect substantial data during each stage through surveys and interviews to measure employees' satisfaction. These data are usually analysed and processed manually. However, employees and trainees tend to share their honest opinions and thoughts over different channels, such as social media and websites. They may anonymously or publicly share positive or negative employer experiences. As a result, the company brand is affected by employees' sentiments. Being seen in a negative light as an employer makes it difficult for a company to attract, recruit or retain talented employees.

This research will use machine learning and neural network (NN) models to classify and predict candidate sentiments towards their job interview. Measuring and classifying the candidates' experiences will help corporate management locate issues and problems within their hiring strategy and give them the opportunity to enhance their onboarding experience and refine their recruiting strategy. As a result, it will increase the satisfaction of future employees and reduce the attrition rate in the company by hiring the best-suited candidates.

II. Related Work

A variety of machine leaning classification models were discussed in the literature and were used for sentiment classification. The study by [3] was used as a basis for many studies related to sentiment classification. The authors investigated the use of machine learning to classify movie reviews. They trained and evaluated Naïve Bayes, maximum entropy classification and support vector machines and were able to reach 82.9% accuracy using SVM and unigram. The authors in [4] used movie reviews to compare the performance of NB and SVM classifiers. They also used CountVectorizer and term frequency–inverse document frequency (TF–IDF) algorithms to convert text data to numerical vectors and reached 94% accuracy with an SVM classifier. [5] used 1,940 customer reviews of a product to train and evaluate an SVM classification model. They achieved an 78% accuracy rate. In [6], the researchers developed machine learning models and lexicon techniques that performed sentiment analysis on employee comments on the Kununu career website. They were able to develop a support vectors machine (SVM) model that scored 79.17 F-measure. In addition, they developed Naïve Bayes (NB) classifier and SentiStrength lexicon models that achieved 72.92 and 75.86 F-measure respectively. [7] developed several deep learning models to conduct employee sentiment analysis using data from Glassdoor.com. The researchers were able to retrieve 1,015,163 reviews for 4,183 companies and evaluated the performance of N-ReLU and bidirectional LSTM. They reached 46.4% accuracy using bidirectional LSTM with GloVe.

In recent years, neural network techniques have shown promising results in the field of sentiment classification.

Convolutional neural network (CNN) and recurrent neural network (RNN) are popular neural network models and have both been used for sentiment classification in the past. The researchers in [8] implemented a series of experiments using CNN. The developed CNN was trained on top of pre-trained word2vec word embeddings and was given seven different tasks, including sentiment analysis and question classification. The researchers compared their results with related work, and CNN was able to outperform on four of the seven tasks. Recurrent convolutional neural network was introduced in [9], where the authors proposed the use of a recurrent structure to capture contextual information and a convolutional neural network for text representation learning. The results showed a slight improvement compared to other techniques used on the same dataset. In addition, a novel model called C-LSTM proposed by [10] uses CNN to retrieve the sequence of high level phrases and pass the results to long short-term memory RNN (LSTM-RNN) to compute the sentence representation. C-LSTM produced promising results.

III. PROPOSED METHODS

In this research, we have explored two approaches. The first approach involved the use of several machine learning algorithms in the classification of interview experiences. In this approach, several feature representations and weighting techniques are used. Each classifier will be trained in the following weighting approaches: term frequency, term presence and term frequency–inverse document frequency. Also, each classifier will be trained using different n-gram values. The second approach involved the use of long, short-term memory (LSTM) neural networks with two word-embedding techniques (skip-gram and continuous bag-of-words).

A. Data Pre-Processing

The data pre-processing step is crucial before developing ML and NN classifiers. In general, the data set was tokenised, converted to lower case and punctuations were removed. Furthermore, The pre-processing techniques are split into three types:

- Type A: The dataset was stemmed and standard stop words were removed. In this research, Type A is referred to as (SSW).
- Type B: The dataset was stemmed and custom stop words were removed. Type B techniques are referred to as (CSW).
- Type C: The dataset was not stemmed and stop words were not removed. Type C techniques are referred to as (NSSW).

The standard stop words list includes 190 stop words, such as "also", "isn't", "not" and "aren't". We also introduced a customised list of 169 stop words. The new list is a subset of the standard stop words, excluding negation words such as "isn't" and "aren't". Negation words such as "not" and "didn't" have a significant effect on subsequent words and can affect the overall sentiments

B. Feature Representation

Documents are a sequence of text of different lengths. Such data need to be represented as features to help perform further calculations. The following sections introduce some of the words representations and weighting methods used in this research.

- N-gram: The n-gram represent a sequence of terms in a document. The value of n represents the number of terms linked together. If n=1, it is a unigram and only one term is used, such as "the", "phone" or "interview". If n=2, it is a bigram and two terms are joined, such as "the phone" or "phone interview" [11].

C. Feature Weighting

Let us assume we have a set of documents where $D = \{d^1, d^2, d^3, \dots d^n\}$ and a set of terms where $T = \{t_1, t_2, t_3, \dots t_m\}$. To classify the documents to the corresponding class from a set of classes $C = \{c_1, c_2, c, \dots c_c\}$, the documents must first be converted to a computable pattern. Below, we explore several methods by which documents can be converted to a weighted vector. Each weight w_k^i represents the importance of each term t_k in the document d^i [12].

$$d^i \rightarrow w^i = (w_1^i, w_2^i, \dots w_T^i)^T \in R$$

- Term Binary (TB): also called term presence and is considered to be the simplest way to convert textual features to vectors. It checks for the existence of a term in a document and returns 1 if the term exists and the term frequency is > 0, or 0 if it does not exist in the document [13].
- Term frequency: returns a vector of the number of times term t_k was used in document
- Term Frequency–Inverse Document Frequency (TF–IDF): proposed by [14]. It is used to calculate the weight of term t_k in the document d^i using the following formula:

  $$w_k^i = tf_k \times \log(\frac{N}{df_k})$$

  Where tf_k is the frequency of term t_k in document d^i, N is the total number of documents |D| and df_k is the number of documents containing the term t_k [15]. This approach will rank words by their importance.

D. Classification

Several machine learning classifiers were used to classify the candidates' sentiments after a job interview.

SVM is a supervised machine learning model [16]; it is a non-probabilistic model that can be used to solve both classification and regression problems. SVM distinguishes between classes using a decision boundary, referred to as a hyperplane. Each data point is classified as either {1,-1} based on the position of the data point in relation to the hyperplane

Logistic regression is a popular supervised machine learning model used for classification tasks. Spam classification for emails is a common application of logistic regression with text. Logistic regression is a binary classifier that calculates the probability function of the output hypothesis h_θ [16].

Naïve Bayes is a supervised learning algorithm based on probabilistic measures that uses Bayes' theorem [17] with a "naïve" assumption to indicate that all features are independent of each other. The posterior probability P(x|y) of an event *x* accruing after seeing data *y* can be calculated as follows [16]:

$$P(x|\boldsymbol{y}) = \frac{P(\boldsymbol{y}|x)P(x)}{P(\boldsymbol{y})}$$

Random forest is a powerful supervised machine learning model and is referred to as an ensemble learning model that trains the data using multiple decision trees. Final classification results are based on the number of votes collected from all the decision trees [18] [19].

IV. DATASET AND TOOLS

In this research, we retrieved 3,983 comments posted in Glassdoor[1] describing the overall interview experience of Amazon hiring process. The dataset includes 2,000 positive experiences and 1,983 negative experiences.

The following comment is an example of a positive experience: "*Recruiter was very responsive and pleasant. Great communication throughout the entire process. The interview onsite was very relaxed and I got a great feel for the work environment. Extremely nice people and smart….*"

The following comment describes a negative experience: "*The interview process was a very long drawn out... The staffing service was slow and did not do a good job expressing the job duties and the nature of the job.*"

For this study, MATLAB R2018a was used to develop machine learning and recurrent neural network classifiers.

V. EXPERIMENTS

A. *Machine Learning Classification*

This section discusses and illustrates the use of machine learning classifiers for sentiment classification tasks using the following ML models to classify interview experiences: SVM, NB, logistic regression, random forest and KNN (K=3). Each classifier was trained with the following weighting methods: TF, TP, and TF-IDF. Each weighting method was experimented using unigram (SSW), bigram (CSW) and unigram bigram trigram (CSW). In addition, all models were tested and evaluated using five-fold cross-validation. Figure 1 illustrates the methodology used for training and developing machine learning classifiers. The following subsections present the results of each experiment based on the weighting approaches.

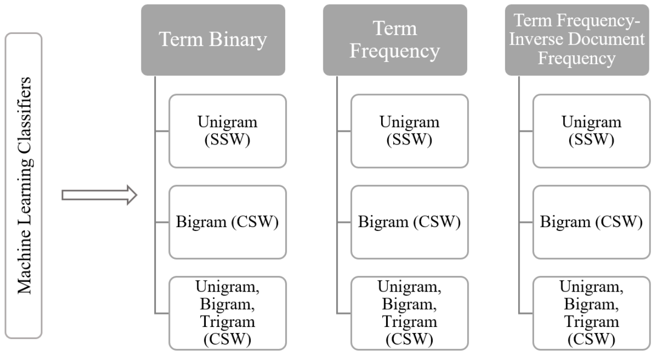

Figure 1. Methodology used to develop machine learning classifiers

1) *Term Binary*

In this section, the term binary weighting approach was applied. Each machine learning model was trained using a term binary model and SSW pre-processing approach. Initial experiments were conducted in unigram representation. Table I illustrates the results of each ML classifier with their precision, recall, accuracy and F1-measure based on unigram. In this experiment, Naïve Bayes scored the highest F1 score with 0.813, while KNN scored 0.621 F1-measure with k=3. Several K values were tested with KNN and it was found that K=3 performed better than other k values. However, KNN is an unsupervised classifier and is considered a lazy learner. Hence, it scored lower than the other classifiers.

TABLE I. CLASSIFIER PERFORMANCE WITH TERM BINARY UNIGRAM AND SSW

Model Type	Precision	Recall	Accuracy	F1-Measure
SVM	0.730	0.748	0.734	0.739
Logistic Regression	0.738	0.842	0.771	0.786
Naïve Bayes	0.753	0.885	0.796	**0.813**
Random Forest	0.727	0.803	0.749	0.763
KNN = 3	0.531	0.747	0.542	0.621

a. Bold values indicate highest F1 score

A second experiment was performed on the machine learning classifiers, using bigram feature representation and CSW. Table II shows the results of that experiment: Naïve Bayes scored the highest with 0.742 F1-measure but scored lower when compared to the previous experiment. SVM and logistic regression scored very close to NB. Table III illustrates the scores of machine learning models that experimented with unigram, bigram and trigram at the same time. Naïve Bayes scored the highest with 0.785. However, the performance of SVM, logistic regression and random forest increased significantly.

[1] www.glassdoor.com

TABLE II. CLASSIFIER PERFORMANCE WITH TERM BINARY BIGRAM AND CSW

Model Type	Precision	Recall	Accuracy	F1-Measure
SVM	0.689	0.776	0.711	**0.730**
Logistic Regression	0.710	0.761	0.724	**0.735**
Naïve Bayes	0.619	0.927	0.677	**0.742**
Random Forest	0.580	0.876	0.619	0.698
KNN	0.505	0.822	0.506	0.626

a. Bold values indicate highest F1 score

TABLE III. CLASSIFIER PERFORMANCE WITH TERM BINARY WITH UNIGRAM, BIGRAM, TRIGRAM AND CSW

Model Type	Precision	Recall	Accuracy	F1-Measure
SVM	0.741	0.807	0.761	**0.772**
Logistic Regression	0.731	0.794	0.750	0.761
Naïve Bayes	0.671	0.947	0.740	**0.785**
Random Forest	0.725	0.803	0.748	0.762
KNN	0.515	0.685	0.518	0.588

a. Bold values indicate highest F1 score

2) Term Frequency

This section explores the effect of the term frequency approach on the classifier's performance. The experimental structure is similar to that of the previous section. Table IV illustrates the results of term frequency with unigram representation and SSW. The Naïve Bayes F1-measure scored the highest, with 0.812. The scores for all models were very close to term binary unigram in Table I.

TABLE IV. CLASSIFIER PERFORMANCE WITH TERM FREQUENCY UNIGRAM AND SSW

Model Type	Precision	Recall	Accuracy	F1-Measure
SVM	0.735	0.710	0.726	0.722
Logistic Regression	0.710	0.822	0.742	0.762
Naïve Bayes	0.751	0.883	0.794	**0.812**
Random Forest	0.742	0.813	0.764	0.776
KNN	0.537	0.767	0.551	0.632

a. Bold values indicate highest F1 score

Table V demonstrates the classifier's performance after implementing the term frequency with bigram and CSW. Similarly, Naïve Bayes scored highest in this experiment but lower than the term frequency unigram noted in Table IV. However, random forest performance increased to 0.745 when compared to the term binary bigram experiment in Table I. Table VI shows the results of term frequency using unigram, bigram and trigram with CSW pre-processing. Naïve Bayes scored 0.802, higher than the term binary with the unigram, bigram, trigram results of Table III.

1) Term Frequency–Inverse Document Frequency

This section uses TF–IDF with the interview experience dataset. The first experiment uses unigram and SSW, and the results of this experiment are shown in Table VII. In this experiment, logistic regression scored the highest with 0.803 F1-measure and Naïve Bayes performance decreased to 0.793.

TABLE V. CLASSIFIER PERFORMANCE WITH TERM FREQUENCY BIGRAM AND CSW

Model Type	Precision	Recall	Accuracy	F1-Measure
SVM	0.690	0.775	0.712	0.730
Logistic Regression	0.720	0.733	0.723	0.726
Naïve Bayes	0.626	0.938	0.687	**0.751**
Random Forest	0.642	0.889	0.695	**0.745**
KNN	0.510	0.976	0.516	0.670

a. Bold values indicate highest F1 score

TABLE VI. CLASSIFIER PERFORMANCE WITH TERM FREQUENCY AND UNIGRAM, BIGRAM, TRIGRAM AND CSW

Model Type	Precision	Recall	Accuracy	F1-Measure
SVM	0.736	0.751	0.739	0.743
Logistic Regression	0.731	0.832	0.762	0.778
Naïve Bayes	0.774	0.834	0.794	**0.802**
Random Forest	0.740	0.803	0.759	0.770
KNN	0.548	0.591	0.549	0.568

a. Bold values indicate highest F1 score

TABLE VII. CLASSIFIER PERFORMANCE WITH TF–IDF UNIGRAM AND SSW

Model Type	Precision	Recall	Accuracy	F1-Measure
SVM	0.701	0.737	0.710	0.719
Logistic Regression	0.760	0.851	0.790	**0.803**
Naïve Bayes	0.751	0.841	0.780	**0.793**
Random Forest	0.721	0.783	0.739	0.751
KNN	0.544	0.807	0.563	0.650

a. Bold values indicate highest F1 score

However, when all classifiers were trained with bigram representation and CSW the logistic regression performance decreased, as shown in Table VIII. Logistic regression outperformed the other classifiers. Table IX illustrates the results of using TF–IDF with unigram, bigram, trigram. Logistic regression scored the highest F1-measure, with 0.814.

TABLE VIII. CLASSIFIER PERFORMANCE WITH TF–IDF BIGRAM AND CSW

Model Type	Precision	Recall	Accuracy	F1-Measure
SVM	0.629	0.841	0.671	0.720
Logistic Regression	0.727	0.830	0.758	**0.775**
Naïve Bayes	0.741	0.730	0.736	0.735
Random Forest	0.642	0.889	0.695	0.745
KNN	0.502	0.999	0.503	0.669

a. Bold values indicate highest F1 score

B. LSTM Classification

This section looks at LSTM with sentiment classification tasks. LSTM can handle sequence data with long dependence, such as text. Four experiments were developed using the word-embedding techniques CBOW and skip-gram. We also explore the effect of data pre-processing on LSTM performance with each word-embedding approach.

1) Word2vec Embedding

Before training LSTM, it is necessary to convert sequences of words into vectors of computable numbers. In recent years, word2vec algorithms have delivered very promising results representing words as vectors using neural networks, and they can learn the relationships between words. In this research, skip-gram and CBOW were used as the word-embedding scheme. The results of these algorithms can be clustered and represented in a t-SNE plot [20]. Figure 2 shows a subset of word representation clustered based on shared semantics found using word2vec, specifically "code" cluster which lists all associated words, such as program, java, data, SQL, etc.

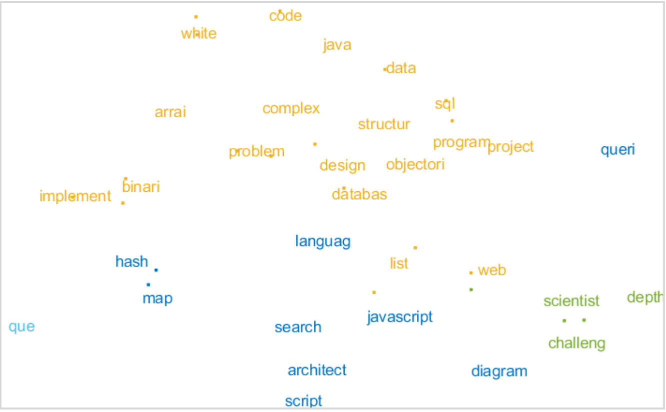

Figure 2. A closer look at the "code" cluster

2) LSTM Classification

In this research, a single LSTM network with 180 units was developed. Different numbers of units were tested, starting from as low as 2 and increasing to 180 units. It was found that 180 units were suitable for the current data set. Figure 3 visualise the LSTM network used in this research [21]. A sequence input layer is used to input a sequence of data into the network. Followed by an LSTM and a fully connected layers . Finally, results will be processed with a softmax layer and passed to a classification output layer to generate the final class label.

Table X illustrates the performance of the four experiments. First, LSTM with CBOW and NSSW pre-processing were implemented: the dataset was tokenised and punctuation was removed. As a result, this approach achieved 0.738 F1-measure. However, when stemming was applied and custom stop words were removed, i.e. using the CSW approach, the performance of LSTM with CBOW increased significantly to 0.776 F1-measure. The third and fourth experiments used skip-gram as word embedding for LSTM with NSSW and CSW respectively. The results were very close, such that skip-gram with NSSW scored 0.756 and skip-gram with CSW scored 0.745.

Figure 3. LSTM network [21]

TABLE IX. CLASSIFIER PERFORMANCE WITH TF–IDF AND UNIGRAM, BIGRAM, TRIGRAM AND CSW

Model Type	Precision	Recall	Accuracy	F1-Measure
SVM	0.707	0.840	0.745	0.768
Logistic Regression	0.775	0.857	0.803	**0.814**
Naïve Bayes	0.775	0.782	0.776	0.778
Random Forest	0.734	0.799	0.753	0.765
KNN	0.498	0.820	0.494	0.619

a. Bold values indicate highest F1 score

TABLE X. LSTM CLASSIFICATION RESULTS FOR INTERVIEW EXPERIENCE

Embedding Word Model	Precision	Recall	Accuracy	F1 Measure
CBOW & NSSW	0.744	0.732	0.741	0.738
CBOW & CSW	0.756	0.798	0.771	**0.776**
Skip-gram & NSSW	0.745	0.768	0.754	0.756
Skip-gram & CSW	0.776	0.717	0.756	0.745

a. Bold values indicate highest F1 score

C. *Misclassification Analysis*

This section discusses and analyses cases misclassified by the classifiers. It also explores examples of misclassification. As logistic regression outperformed the other machine learning classifiers, the focus will be on analysing logistic regression performance.

Logistic regression scored a 0.814 F1 score. As a result, 785 out of 3,983 records were misclassified, of which a total of 286 positive experience interviews were wrongly misclassified as negative. In addition, 499 negative experiences were classified as positive. After examining the misclassified records, it was possible to identify the causes of misclassification.

A subset of the misclassified comments did not include sentiment words: either the author of the comment described the process in general or listed all the questions that were asked during the interview. Below is an example of positive experience that was classified as negative.

"I was invited for a phone interview after applying online, started off with general behavioural question and then moved to test my knowledge on finance. The interview was divided into 30 mins behavioural and 10 mins technical with remaining five mins for any questions."

A second reason for misclassification was the use of negative terms when describing an experience classified positively by the authors. As a result, the logistic regression classifier classifies the experience as negative. Below is an example of a positive experience comment that was classified as negative.

"Had three phone interviews, … No tricky questions. I declined, because they were… extremely vague on pay raises and stock raises. The package isn't that great either: Benefits average… In addition, I have heard mostly negative feedback from working at Amazon (2 year average retention rate), and not having a clear idea for pay raises shows that they would have no respect for me in this position."

Similarly, negative comments were misclassified as positive due to the use of positive words within the comment. Below are examples of misclassified negative experiences.

"Good process, fast, and efficient. Interviewed by a total of 8 individuals over the course of a about 10 days. Received an in-person interview in Seattle. My advice would be to dress casual, be yourself, and know their core principles by heart."

"I got an email from the recruiter from Amazon to setup phone interview…. The interviewer was friendly. Overall good experience but I did not do very well probably I had not prepared enough. He asked programming and testing questions."

VI. CONCLUSION

Job interviews are critical to secure talented employees at any company. A good hiring strategy can attract the right candidates to the right position. However, a negative onboarding process can impact the company brand and discourage talented candidates from accepting a job offer. Machine learning and neural networks can help companies towards a better understanding of their employees, trainees and future candidates. The main objective of this research is to use machine learning and neural network algorithms to predict the results of job interviews and candidates' overall sentiment experience. Several experiments were developed to examine the role of machine learning and neural networks in predicting candidates' sentiments towards the employer. It was found that logistic regression with TF–IDF and unigram word representation outperformed all trained machine learning classifiers, with 0.814 F1-measure.

The experiments conducted in this study revealed several outcomes that answered the research questions. First, it was found that the selection of pre-processing and word representation schemes has an effect on classifier performance: the use of bigram decreases classifier performance, while unigram or unigram, bigram and trigram have a positive effect on classifier performance. Secondly, it was found that the choice of feature weighting affects performance, and TF–IDF performed performs very well for the classification tasks. Finally, experiments revealed that the performance of LSTM networks using CBOW were significantly enhanced when the dataset was steamed and stop words were removed.

VII. REFERENCES

[1] J. Morgan, The Employee Experience Advantage: How to Win the War for Talent by Giving Employees the Workspaces They Want, the Tools They Need, and a Culture They Can Celebrate, John Wiley & Sons, 2017.

[2] E. Lupfe, "The employee lifecycle is your roadmap to building an engaged employee experience," The Social Workplace, 15 November 2017. [Online]. Available: http://thesocialworkplace.com/2017/11/the-employee-lifecycle-is-your-roadmap-to-building-an-engaged-employee-experience/. [Accessed 23 July 2018].

[3] C.-F. Chien and L.-F. Chen, "Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry," Expert Systems With Applications, vol. 1, no. 34, p. 280–290, 2008.

[4] A. Tripathy, A. Agrawal and S. K. Rath, "Classification of sentimental reviews using machine learning techniques," Procedia Computer Science, vol. 57, p. 821–829, 2015.

[5] C. Bhadane, H. Dalal and H. Doshi, "Sentiment analysis: Measuring opinions," Procedia Computer Science, vol. 45, p. 808–814, 2015.

[6] J. Abel, K. Klohs, H. Lehmann and B. Lantow, "Sentiment-analysis for German employer reviews," in International Conference on Business Information Systems, 2017.

[7] T. E. Whittle, "The Stanford NLP Group," 2018. [Online]. Available: http://web.stanford.edu/class/cs224n/reports/6880837.pdf. [Accessed 15 Jun 2018].

[8] Y. Kim, "Convolutional neural networks for sentence classification," in EMNLP Conference Proceedings, 2014.

[9] S. Lai, L. Xu, K. Liu and J. Zhao, "Recurrent convolutional neural networks for text classification," Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, vol. 333, p. 2267–2273, 2015.

[10] C. Zhou, C. Sun, Z. Liu and F. Lau, "A C-LSTM neural network for text classification," arXiv preprint , vol. arXiv:1511.08630, 2015.

[11] D. Jurafsky and J. H. Martin, Speech and Language Processing, London: Pearson, 2014.

[12] F. R. López, H. Jiménez-Salazar and D. Pinto, "A competitive term selection method for information retrieval," in International Conference on Intelligent Text Processing and Computational Linguistics, Berlin, Heidelberg, 2007.

[13] G. Paltoglou and M. Thelwall, "A study of information retrieval weighting schemes for sentiment analysis," in Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 2010.

[14] G. Salton, A. Wong and C.-S. Yang, "A vector space model for automatic indexing," Communications of the ACM, vol. 18, no. 11, p. 613–620, 1975.

[15] V. B. Vaghela and B. M. Jadav, "Analysis of various sentiment classification techniques," Analysis, vol. 140, no. 3, p. 22–27, 2016.

[16] S. Rogers and M. Girolami, A first course in machine learning, CRC Press, 2016.

[17] T. Bayes, R. Price and J. Canton, "An essay towards solving a problem in the doctrine of chances," Reprinted in Facsimiles of Two Papers by Bayes, vol. 53, 1963

[18] T. K. Ho, "Random Decision Forests," in Proceedings of the Third International Conference on Document Analysis and Recognition, 1995.

[19] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, p. 5–32, 2001.

[20] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," Journal of Machine Learning Research, vol. 9, p. 2579–2605, 2008.

[21] "LongShort-TermMemoryNetworks,"MathWorks, [Online].Available:https://www.mathworks.com/help/nnet/ug/long-short-term-memory-networks.html. [Accessed 01 September 2018].