NOVA IMS Information Management School

MEGI

Mestrado em Estatística e Gestão de Informação
Master Program in Statistics and Information Management

SME CREDIT APPLICATION

A TEXT CLASSIFICATION APPROACH

Daniela Saavedra López

Project Work presented as partial requirement for obtaining the Master's degree in Statistics and Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

SME CREDIT APPLICATION, A TEXT CLASSIFICATION APPROACH

by
Daniela Saavedra López

Project Work presented as partial requirement for obtaining the Master's degree in Statistics and Information Management, with a specialization in Market Research and CRM

Advisor: *Professor Mauro Castelli*

June 2020

ABSTRACT

During the SME credit application process a credit expert will give a specific recommendation to the credit commercial advisor. This recommendation can be classified as positive, negative or partial. This project aims to construct a text classifier model in order to give the recommendation text one of the categories mentioned before. To achieve this, two models are tested using state-of-the-art architecture called BERT proposed by Google in 2019.

The first model will use single sentence BERT classification model as proposed by Google. The second model will use SBERT architecture, where BERT embedding model will be fine-tuned for the specific task, a max-pooling layer is added to extract a fixed size vector for all the document and work under fully connected network architecture. Results show that the second approach got better results regarding accuracy, precision and recall. Despite of the bunch of limitations of computational capacity, limited number of tagged examples and BERT maximum sequence length the model show a good first approach to solve the current problem.

KEYWORDS

Natural Language Processing (NLP); Banking; Credit application; Small and medium enterprise (SME); Neural Networks (NN); Bi-directional Encoder Representations for Transformers (BERT)

INDEX

TABLES INDEX

EQUATIONS INDEX

LIST OF ABBREVIATIONS AND ACRONYMS

SME	Small and medium enterprise
NLP	Natural Language Processing
NN	Neural Networks
ELMo	Embeddings from Language Models
BERT	Bidirectional Encoder Representations from Transformers
SBERT	Sentence-BERT
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory

1. INTRODUCTION

1.1. CONTEXT AND PROBLEM DEFINITION

This project aims to solve a current problem a bank has when analyzing and classifying texts generated during the credit application process of Small and Medium Enterprises (SME) clients. Banks credit processes vary according to the product and the type of client as each process takes different inputs for decision making. So, when a SME applies for a credit, the bank manager creates a new request which should specify the products and conditions of the credit the client is asking for. Then, each credit request goes directly to a study stage where a risk analyst determines the internal and external credit behavior of the client, the economic status of the SME and the company's financial statements to evaluate how solid it is. At that time, the risk expert issues a concept explaining if he partially[1], fully agree or disagree with the credit request and its further approval. At the end, the bank manager, in correspondence with the powers of the office, approves or not the credit application.

Making an analysis over the conditions where a credit product was granted to a client, the credit risk area realized they had no way to see if the bank managers did follow or not their recommendations about approving or not a credit application. To solve this problem, texts should be analyzed and classified in one of three possible categories, thus the credit risk area may quantify the number of recommendations the commercial managers followed and the relationship between those references and the existing past-due portfolios. This is a very important problem for the risk area to solve, first because no longer projects have worked with text analysis and classification and second because a transversal analysis is needed to determine if the credit application process is minimizing the credit risk.

This development has as references the following: Spanish recommendation texts are transformed into numeric vectors as input for a classification model. Preprocessing process should be done even when working with non-structured data, in this case documents had to be normalized, lower case was applied and, as a consequence, numbers, stop words and regular expressions were removed. Having a normalized corpus, document embedding methods were performed: first, BERT single sentence classifier pre-trained model was mixed with BERT multi-lingual based embedding model. In the second approach, BERT algorithm was fine-tuned with training texts and pooling layer helped to reduce dimensionality to finally get a 768-dimension vector for each recommendation so later a neural network is performed to classify texts. With the texts classified, the company has a descriptive analysis of how the area of experts has issued the concepts over time; a comparative analysis of what the expert recommended and what the bank manager decided and, finally, more variables can be added to calculate the impact of following or not the recommendations over the past-due portfolio. Here is the aggregated value because companies work for better understanding of clients and internal processes. Without this kind of analysis, the bank is exposed to an operational risk defined as the possibility of loss resulting from inadequate or failed internal processes or from external events. This definition includes legal risk but excludes strategic and reputational risk (Pakhchanyan, 2016). Instead, if risk is avoided from the first moment, it will lead to make more accurate decisions about which client the bank should lend a loan. These decisions have a huge impact in terms of costs and minimize the

[1] A partial agreement means the expert approves the transaction but suggest a change in the amount, the interest rate, time period of the credit or ask to change the collateral.

negative effect over economic value of the bank itself, because a bank economic value and risk ranking depends on how risky its clients are.

As mentioned before, the general goal of this study is to classify loan study recommendations into three different categories. For this, NLP techniques will be used to transform text into numbers for further use in a classification method, in this case artificial neural networks work behind it.

According to this, credit risk analytics group defines if the bank manager adopted the recommendations and how tight, to conclude whether if:

1. Bank manager did follow the advice and the client resulted to be a good client. So, study loan area should maintain its policies.
2. Bank manager did follow the advice and the client resulted to be a bad client. So, policies inside the bank should be modified because either it shows managers know better the client or when writing recommendations experts are missing important information.
3. Bank manager did not follow the recommendation because he thought to knew better his client and this resulted to be good.
4. Bank manager did not follow the recommendation and the client did not resulted to be a good one, so they do not have instinct to read when they are facing a bad client, or they just approve loans just to get a commission.

To achieve the overall goal of the study it is necessary to solve the following items on the way:

1. Normalize text by removing stop words, numbers, accent marks and change into lower case.
2. Compare if fine-tuned BERT document vector representation model performs better than BERT pre-trained model.
3. Compare classification performance between single sentence BERT classifier or fine-tuned BERT followed by a pooling layer.

All three specific objectives depend one on another as they stablish the roadmap to achieve the overall goal. The first one refers to the transformation and normalization of the texts to reduce as much as possible noisy words. Then, normalized text needs to be transformed into numerical vectors and two options are presented, either use BERT multi-language pre-trained word embedding model and train BERT single sentence classifier or fine-tune BERT pre-trained model and add a pooling layer to extract a fixed length vector for each recommendation and through a fully connected neural network classify texts into positive, negative or partial.

This report shows first a literature review of word numerical representations used in Natural Language Processing and classification algorithms for text. After that, the methodology and process description are presented followed by the results obtained, discussion and conclusions. Finally, limitations and recommendations for future versions of the model are exposed.

2. LITERATURE REVIEW

The challenge to be solve dives into the text classification task. To achieve this goal, literature review over document representation in numerical vector space and classification algorithms for texts had to be done to analyze advantages and disadvantages of each possible approach.

2.1. NATURAL LANGUAGE PROCESSING (NLP)

Natural language processing (NLP) is a theory-motivated range of computational techniques for the automatic analysis and representation of human language (Young, Hazarika, Poria, & Cambria, 2018). Representations focus on the understanding of the structure and meaning of written and spoken communication and comprehends a huge scope of tasks. These tasks include common word sequences identification, Part-of-Speech tagging, association between words and other more complex such as classifying texts, summarizing, solve question answering, speech recognition, machine translation and creation of conversational dialog systems (Cambria, Poria, Gelbukh, & Thelwall, 2017).

This project relies on text classification task, as for each recommendation text should be assigned one of the three predefined categories. This task will be developed using several NLP techniques for the representation of the whole document in a numerical vector that is the input for the classification algorithm. To have a cozier approach to this NLP task we can analyze the popular sentiment analysis where texts, usually paragraphs or sentences are labeled into predefined categories such as happiness, anger or sadness. Usually companies use sentiment analysis to get insights about their online product reviews, social media comments from posts in Twitter or Facebook to understand how clients feel towards the brand. Stock markets, elections, disasters, medicine, software engineering and cyberbullying (Mäntylä, Graziotin, & Kuutila, 2018) are some of the topics and industries where sentiment analysis has been successful and has given important and tangible insights as branch of text classification models.

2.2. TEXT NUMERICAL REPRESENTATION

Text classification has evolved through years. At first, representations of words were made by one-hot encoders where the text was reduced to a dichotomous vector that expressed if word was part of the text or not. Then, Bag-of-Words came into the field and this representation technique only include information about the terms and their corresponding frequencies in a document independent of their locations in the sentence or document (Altınel & Ganiz, 2018) and the problem lied on the lack of semantic and syntactic understanding of the text itself so at the end words with multiple meanings are treated as a unique word. In synthesis, Bag-of-Words creates a VxM matrix where 'V' represents the dimension of the vocabulary of the texts and 'M' takes the examples dimension, having this the matrix is a binary representation if the word $v_1$is present in the example m_1. Due to Bag-of-Words weaknesses, approaches to semantic representation of a text was the next challenge to be faced and Google Team solved it in 2013 with the publication of two new models, the Continuous Bag-of-Words Model (CBOW) and Continuous Skip-gram Model where neural networks helped to reach the state-of-the-art performance in word representation of vector space for measuring syntactic and semantic word similarities (Mikolov, Chen, Corrado, & Dean, 2013). CBOW and Skip-gram are neural network based models where n-gram window is used to predict the current word due to its context in CBOW

model and Skip-gram tries to maximize the classification of a word based on other word in the same sentence (Mikolov et al., 2013). In addition, some authors focused their work on character embedding that meant concentrating not on the word itself, but on the place of the letters in relation to each other. Unfortunately in this case, very opposite words may be represented in the same vector space, because they only differ in prefixes, so they are considered orthographically similar. Even though research had been focused on word vectors, authors went beyond and began to explore sentence and document representation in vector space which, until that moment, was adopted as the average vector for the document word embedding. This solution did not take into account the order of words in the text, so the paragraph vector was proposed by Google in 2014 when Mikolov and Le presented an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts, such as sentences, paragraphs and documents (Mikolov, Tomas ; Le, 2014). The results were catalogued as the new state-of-the-art over text classification and sentiment analysis tasks (Mikolov, Tomas ; Le, 2014).

In semantic text dimensions, it was stated that semantic text classification algorithms achieve better classification accuracies than traditional ones (Altınel & Ganiz, 2018). These two authors present five approaches to semantic text classification where knowledge-based, corpus-based approaches and deep-learning based approaches are the most widely used. Knowledge-based train model over specific domain texts, this approach has been successful in many fields for example over biomedical NLP (Wang et al., 2018). Knowledge-based solution is the opposite of corpus-based approach which are models trained over general text and are not concentrated on a specific topic, for example some of them are trained over huge amount of texts extracted from Wikipedia. Finally, deep learning was tagged by Altınel and Ganiz as the best-in-class performance in classification field but they demand huge computational costs and perform better when having vast amouts of data. Eventhough, there are multiple ways to work on semantic text classification, the authors also highlight some of the challenges that actually are present in this project: 1) availability of a knowledge base for a specific language, 2) processing complexity of a large external knowledge base, 3) complexity of computations to extract latent semantics and 4) computational hardware systems (Altınel & Ganiz, 2018).

Construct a robust embedding model to face a specific problem wil encompass all the challeges mentioned by Altınel & Ganiz (2018). AI teams from Standford, Facebook and Google have developed pre-trained word embedding models and publish them as python packages and frameworks, so the community may use them in their daily tasks throughout transfer learning. Transfer learning can be understood as a high-performance learner used to improve a learner competences from one domain by transferring information from a related domain (Weiss, Khoshgoftaar, & Wang, 2016). It has been used in multiple tasks such as sentiment analysis like presented by R. Liu et al., and also in other fields besides NLP such as recommendation systems presented by Pan (2016) and visual recognition from Zhang, Li, Ogunbona and Xu (2019). In this line of work, deep learning approach to semantic text classification requires a huge amount of data, so the principal obstacles for many researchers to use transfer learning is the lack of data to train a well-performed model that will capture all semantic generalizations and specificity of the domain of the problem to solve. Some of the pre-trained models are FastText, ELMO and BERT. FastText developed by the Facebook AI Research team (2019) presented pre-trained word embeddings for 157 languages using skip-gram and CBOW models trained over Wikipedia and common crawl project data. Evaluation metrics of NLP tasks vary depending on n-gram

length, number of negative samples and number of epochs, eventhoug one of the major contributions was training word vectors in multiple languages on large scale noisy data from the web (Grave, Bojanowski, Gupta, Joulin, & Mikolov, 2019). In complement, Embeddings from Language Models (ELMo) developed by the Allen Institute of Artificial Intelligence and the University of Washington, is a model where word vectors are learned functions of the internal states of a deep bidirectional language model (biLM) (Peters et al., 2018), their methodoly established that in high levels of the Long Short Term Memory (LSTM) NN was captured the context-dependent aspects of the word meaning and in lower-levels the syntax (Peters et al., 2018). However, ELMo results are only available for english so it can no longer be used in this project. In 2019, Google AI Language team used deep bidirectional neural networks to capture all context before and after each word, contrary to ELMo where token representation was a concatenation of independent left-to-right and right-to-left representations (Devlin, Chang, Lee, & Toutanova, 2018). One of the most relevant advantages of BERT is that pre-trained model could be fine tuned with one additional output layer (Devlin et al., 2018) and performed as the new state-of-the-art in many NLP tasks including sentiment analysis. This model was trained over BooksCorpus which had 800M words and Wikipedia text passages (2,500M words) (Devlin et al., 2018) and results are available for 104 languages including spanish.Using BERT pre-trained model for spanish will bring solution to some of the challenges over semantic text classification mentioned by Altınel and Ganiz. First and most important is the availability exclusively for spanish language, then as being trained over a huge dataset of general texts from BookCorpus and Wikipedia its bidirectional neural network structure captures semantic and syntaxis important features from particular spanish language. This means it will be no need to train a model from scratch and will contribute to reduce the computational training time and it can be fine-tuned to adjust parameters to the specific task of the project.

2.3. TEXT CLASSIFICATION

Over years NLP studies have evolved according to the evolution of technology which has allowed that ancient concepts like neural networks could be tested and trained. That means that even though deep learning algorithms fundamental concepts were born in the 1950s' is not before 2010's when they showed their real power when Graphic Processing Units (GPUs) speed increased dramatically and performed in some cases 50 times faster than algorithms trained in standard CPU versions (Schmidhuber, 2015). In 2011, trained NN over GPU was the first system to achieve superhuman vision pattern recognition focused on traffic sign identification which meant an enormous step over self-driving cars developments. Deep learning is defined as a representation-learning method with multiple levels of representation, obtained by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level (Lecun, Bengio, & Hinton, 2015). The neural network based approach have had various advantages as they help to capture the proper syntactic role of each word and also the syntactic and semantic structure of the text (Gupta & Gupta, 2019). As deep learning methods like Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have shown very good results over NLP tasks, specifically for supervised text classification (Alom et al., 2019). Other proposal is Recurrent Convolutional Neural Networks a mix of RNN and CNN, where RNN captures context information from word representation which the conclude may introduce considerably less noise compared to traditional window-based neural networks, in addition they employ a max-pooling layer

(from CNN) that automatically judges which words play key roles in the text classification task (Lai, Xu, Liu, & Zhao, 2015).

CNN are designed to process data in multiple arrays (Lecun et al., 2015). Its architecture considers two types of layers: the convolutional and the pooling. The role of the convolutional layer is to detect local conjunctions of features from the previous layer and the role of the pooling layer is to merge semantically similar features into one (Lecun et al., 2015). These two layers make CNN popular because of their ability to account data that may not be uniformly or systematically formatted, since they are capable of learning features that may be present in different regions. So in linguistic analysis, this is an advantage due to the identification of a sample text no matter where it occurs (Otter, Medina, & Kalita, 2018). Despite the fact that CNN are very popular in image and video, it has shown very good results over text classification task, for example Kim (2014) compared results using CNNs over sentence classification task based on word2vec pre-trained model and concluded that unsupervised pre-training of word vectors constitute important ingredients in deep learning for NLP (Kim, 2014). When exploring CNNs, some articles determined that very deep convolutional neural networks perform better than state-of-the-art text classification public tasks (Conneau, 2017).

Advantages of RNN lead into its own feeding in the same node, this fact impacts NLP tasks as the meaning of linguistic units depends on word order; so, RNN are said to have memory over previous elements. Some authors suggest using bidirectional algorithm as backward dependencies also exist when word meaning depends on the following one (Otter et al., 2018). The mechanism of RNN is biased because it gives higher relevance to recent words, thus it could reduce the effectiveness when it is used to capture semantics of a whole text (Lai et al., 2015). Long-Short Term Memory Neural Networks is a type of a RNN where vanishing problem is solved by adding oblivion doors to prevent the network gives a lot of weight to the first associations of the NN (Ay Karakuş, Talo, Hallaç, & Aydin, 2018). In this form, it retains significant information during the network while irrelevant information can be forgotten. Bi-directional LSTM is an improved version of the LSTM where simultaneously a NN is trained regarding the text from left-to-right and right-to-left so the output works as a concatenation of both layers. Some studies used bidirectional LSTM and two-dimensional max pooling and obtained better results than CNN and RNN models (Hashimi, Hafez, & Mathkour, 2015). The combined model presented better results because it benefits from capturing the contextual information caused by the RNN and with the max-pooling it is possible to choose which contextual facts are more important in the text classification.

As said before, BERT performs actually as the state-of-the-art model for vector representation of words. Its structure is based on the Transformer Model and the Masked Language Model. Transformer model architecture is a simple network architecture founded solely on attention mechanisms (Vaswani et al., 2017) which usually connect encoders and decoders of recurrent or convolutional neural networks. As attention mechanisms are parallelizable tasks the proposed architecture require less time to train (Vaswani et al., 2017). Masked Language Model which mask some of the tokens of the input and the objective is to predict the original ID of the masked word (Devlin et al., 2018). This MLM objective enables the representation to combine both left and right context of the words and this allows the Google team to train a deep bi-directional transformer (Devlin et al., 2018). Taking into account that BERT can be fine-tuned for a specific task, the single sentence BERT model classification

structure is available for the public to use it and regarding its structure based on Transformer and MLM, it is positioned as the state-of-the art model for classification tasks.

After BERT, sentence-BERT (SBERT) was proposed by the Department of Computer Science in the Technical University of Darmsdat in Germany. The proposed structure harnesses BERT model and adds a max-pooling layer to get fixed size sentence embedding vectors (Reimers & Gurevych, 2019). It shows better results common benchmarks, but as structure it should be taken into account.

The review of the state-of-the-art for text classification task will impact text classification model performance as pre-trained embedding models capture semantic and syntactic general features of the language as being trained over millions of general texts regarding different web sources. In contrast, with the limited quantity of texts available for this project the odds of capturing all these features are very low. Then, the decision is whether the fine-tuning of these pre-trained embedding models with a specific domain text helps the classification model achieve better performance metrics or not. For this reason, both experiments will be tested using bidirectional neural networks with BERT to capture semantic and syntaxis important features of Spanish language and the fine-tune BERT model with recommendation texts. Finally, some authors have demonstrated that max-pooling layers will help bi-directional neural networks to identify and perform type-of feature selection from all extracted features and improve classification algorithm performance metrics. As BERT model architecture is bi-directional Transformer based, a max-pooling layer is added in the second experiment (proposed in SBERT) to test if these helps to achieve better accuracy, precision and recall evaluation metrics.

3. METHODOLOGY

3.1. CREDIT PROCESS

SME credit applications can be classified in two main groups: The first refers to the case when customers access a new product and the second has to do with the case when clients already have a credit limit approved and they request the renewal or modification of its conditions. In both cases, the application must be done by contacting either one of the commercial advisors in the bank or the designated agent for a specific SME portfolio.

The first moment of the credit application process occurs when the commercial agent creates a new credit application on the internal system by filling a designed form, which requires specific information of the client as the national number id, financial statements, economic group and some others. Then, a chart must be completed with the amount asked for each product, the loan terms, and collaterals if it is the case. In addition, the commercial agent could make comments or write a short description related to the client and the motivation of the credit application. After filling out the form, the system redirects the credit application to one of the Credit Risk Department analysts in order to start the second stage of the credit process. In this step, every credit application is analyzed and complemented with external information as reports in the National Credit Bureau, macroeconomic stability and forecasts of the industry expansion or contraction. Having this global client profile, the credit risk agent recommends credit conditions, arguing why (s)he either fully agrees to give or not the credit products to the client or approves partially the operation by suggesting a change in the amount, products, loan terms or collaterals. At the end of the credit process, the commercial agent analyzes the recommendation from the credit risk area and decides whether to approve or not the loan and to modify or not the initial conditions. This means that the commercial agent is free to follow or not the recommendation of the credit risk analyst.

3.2. CHALLENGE

Monitoring the information, the timeline and the decisions made along the application process has been a priority challenge for the bank because, in three steps, information captured is stored in structured form. In the first step, information about the client desirable conditions is saved in data bases, so the analysis and conclusions are extracted regarding product demand. Then, for the final step, if the application is approved it will be registered in the disbursement data base. Thus, with these two inputs, it can be known the disparities between the asked product conditions and its disbursement. In summary, the problem arises when the credit risk area wants to analyze the disparities between what the credit risk agent suggested and what the commercial agent approved because information can only be found in the recommendation text written by the credit analyst. The analysis of this non-structured information differs a lot for what it has been done in the credit risk area. As a consequence, the challenge of this project is to get a closer view around the recommendations made by the Risk Department and aims to help a bank to create a text classification model that will assign one of three categories for each recommendation made by the credit expert to the bank agent. To achieve this goal, NLP tasks are going to be used as tools to develop a classification model for these texts.

Traditional text classification models focus on three main topics: feature engineering, feature selection and usage of machine learning algorithms to perform the classification model (Lai et al., 2015). Feature engineering plays an important role into any supervised or unsupervised model because one could not expect good results if one goes to garbage as input for a model. In NLP task, feature engineering is translated as the need to represent texts into a numerical vector space. Fortunately, some of the most popular AI teams have developed pre-trained models for Spanish language and their results can be implemented by using transfer learning. Then, feature selection task rises into the to-do list. Some authors have concluded that a proxy for feature selection is the usage of a max-pooling layer into the neural network due to the ability to select important features that help with the classification task (Howard & Ruder, 2018; Zhou & Qi, 2016). Finally, in traditional exercises, the classification algorithm is performed. In this case, all three tasks are immersed into a unique framework, but each experiment will have its own configuration.

3.3. DATA

The traceability of the credit request process is centralized over internal systems. Unfortunately, the system is out of date so is not possible the direct connection through drivers to extract them automatically from python. With this scenario, all historical information from January 2015 to January 2019 was extracted manually in 10 different text files of 324.013 credit applications throughout the four years' time window. To download data more efficiently the variables of interest were extracted: the recommendation text, the number identification of the credit application and the client's national id.

# Repetitions by case number	Count
1	323097
2	393
3	29
4	4
5	2
7	1
10	1
Total unique cases	**323527**

Table 1: Repetitions by unique case number id

In the course of an initial data cleaning process, it was noticed that some credit applications appeared in the data base more than once. Table 1 shows the number of cases that appeared 1, 2, 3 or even 10 times and to avoid noisy data, duplicated registers were dropped while keeping the last recommendation text by regarding the most recent date. Additionally, 19,6% of the data base was excluded because recommendation text column was empty.

case_num	id	rec	rec_level
1201010584	2700621621	02-Dic-2010 15:33:57;Recomendación agregada por: Frankin Alexander Rios Palacio;Si bien la sociedad a pesar de su corto tiempo de constitución ha logrado sostener una tendencia creciente en su facturación, registra ahora un nulo apalancamiento con el sector financiero y un controlado nivel de endeudamiento general, existen varias consideraciones a destacar. Hasta ahora, el nivel de ingresos operacionales ha sido bastante modesto respecto al endeudamiento financiero propuesto si se tiene en cuenta que el monto solicitado supera un poco más de los dos años de ventas. Evaluando la rentabilidad del negocio a partir de la tendencia de su margen operacional, se aprecia una gran fragilidad en la estructura financiera si se contrasta el margen del último año ($12MM) con el monto de intereses generado en caso de incurrir en la nueva obligación por la cuantía planteada, al respecto de lo cual se debe tener en cuenta que se ha manifestado de manera explícita la intención de manejar bajas utilidades por efectos fiscal. El capital social actual sería bastante modesto respecto al nivel de pasivo en que incurriría, no existe un compromiso importante por parte de los socios que se evidencia en un aporte significativo de recursos frente a la nueva inversión y los resultados obtenidos luego de proyectar el flujo de caja no fueron satisfactorios en todos los años. En razón a lo anterior, la Gerencia de Crédito no recomienda la OE propuesta. Calificación Interna Recomendada AA;	-1
12011033042	2672109195	18-May-2011 13:59:04;Recomendación agregada por: Paula Andrea Jaramillo Jimenez;Empresa constituida desde el año 1.977 y la cual se dedica a la comercialización de textiles para la confección de prendas de vestir, teniendo como principal proveedor a Fabricato quien posee el 75% de las acciones de la sociedad. Si bien la compañía presenta un crecimiento muy dinámico para todos los años, presenta márgenes muy estrechos y decrecientes, cubriendo estos ajustadamente los egresos no operacionales del negocio y limitando la capacidad de cubrimiento de un eventual servicio de la deuda. Además, para el año 2.010 la sociedad profundiza sus pérdidas, ya que se amplió la fuerza de ventas y se castigó la cartera de periodos anteriores, ofreciendo esta última acción poca mejoría en la rotación de la cartera, la cual continúa en un alto número de días. Finalmente, cuenta con un bajo capital social y si bien posee un importante valor en la revalorización patrimonial, esta es consumida de manera considerable por las pérdidas generadas, especialmente la del último año, lo cual ha generado un crítico indicador de endeudamiento neto. De acuerdo a lo anterior, la Gerencia de Crédito recomienda solo la renovación del LME actual por $66MM. La calificación interna recomendada es A de acuerdo a la plantilla.	0

Table 2: Data base structure

Table 2 shows the structure of the initial data base. The column 'case_num' corresponds to the unique number identification for the credit request, 'id' will identify the national number identification of the client and 'rec' contains the recommendation text (in Spanish) where the credit expert analyst exposes all the pros and cons of the client financial situation. Last but not least the analyst gives his positive, negative or partial conclusion about approving the credit products. It is important to clarify that a positive recommendation means the analysis agree with all the proposed conditions of the loan including credit score, amount, collateral and credit period; a negative recommendation stands for a disagreement with the approval of any credit product; finally, a partial recommendation represents that it is feasible to take the risk of approving the requested credit but modifying one or more conditions, so he could suggest approving only a percentage of the requested amount or asking for more collaterals.

As target variable is needed to train the classification model, 1181 recommendation texts were manually labeled so this dataset was used for training, validation and test. This label can be found in the 'rec_label' column where '-1' signifies negative recommendation, '0' indicates partial recommendation and '1' for positive.

One of the weaknesses when using BERT architecture is that it is limited to 512 tokens, that means the maximum sequence length allowed as input is 512 words. By the nature of the dataset the credit risk analyst does not have any restriction when writing down the recommendation. This means some recommendation texts exceed the limited number permitted by BERT. Doubtless, the proposed BERT architecture will extract contextual information from the limited input, but in this case word stops will have to be removed to shorten the text and let more relevant information get into the BERT model. For this task, NLTK Spanish stop words dictionary was used.

3.4. DATA PREPROCESSING

To guarantee the quality of a model output it is crucial to make a conscious preprocessing of the data to ensure you are not giving the algorithm biased, erroneous or noisy information. In this case, data cleaning encompasses lower case transformation and accent marks, special characters, numbers, additional white spaces, punctuation and stop words are removed (Pons, Braun, Hunink, & Kors, 2016).

3.4.1. First Look

The first step to start with the data preprocessing was texts visual analysis to identify special features to be removed. In this previous step, it was noticed that some of those traits had a repetitive structure when writing the recommendation text because credit risk analyst started the document with the sentence *'the credit management area recommendation is* ____*'* so the blank was either the word positive, negative or partial. This means some of the credit experts followed a given structure. In consequence, this pattern addressed 47,8% and they were no object of this study as the classification was made by regular expression identification process. These recommendation texts gave a first approach of the categories distribution in the recommendation texts data set. 49,5% of the 124.351 credit applications finished the recommendation sentence with positive; the second category with major participation was the partial recommendation that represented the 28,4% of the tagged texts by regular expression; finally, the negative recommendation group is formed with 26.527 credit applications that represented the 22,1%. At the end, it remained 135.633 examples from the original data base that had 324.013 which were object of study in this project.

As shown in the illustrative examples in table 2, the recommendation text column ('rec') included at the beginning the register of date and credit risk agents name. In order to avoid noisy information these two items were removed.

3.4.2. Lowercase letter transformation

One of the most current text transformations is to adapt all the text to lowercase letters (Jianqiang & Xiaolin, 2017; Mohammad, 2018). In some cases, it is not a desirable to process depending on the task. For example, when developing a question-answer model the transformation of the full text to lowercase will make the recognition of proper names more difficult to identify because usually they began with uppercase letter followed by lower case. In this project, contrary to the example mentioned before, the transformation of the text to lowercase will bring more advantages and facilities than disadvantages. First, because the origin of the data we are dealing with allows misspelling which will be totally different if for example the origin of the data is a published book or papers that tend to have

meticulous revision of spelling, semantic and syntaxis of the document. Finally, because it is good to avoid the fact that a word repeated twice or more is treated as two or more different words. For example, if the word 'positive' is written as 'Positive' in one text and 'positive' in another different text, we will induce the model to an error as the same word may have different vector representation and further will contribute differently to the model.

3.4.3. Stop-words

Stop-words are words that are repeated constantly and they have no meaning like pronouns, articles and prepositions (Saif, Fernandez, He, & Alani, 2014; Silva & Ribeiro, 2003), in NLP tasks usually stop-words are removed in order to make the processing more easily and the purpose is to give the algorithm a more cleaned and relevant. Removing this kind of words usually does not affect the semantic meaning of the text as important and relevant words remain.
Several AI teams have developed their own repository of stop-words for different languages. In this project the NLTK package set of stop-words corpus for Spanish language was used.

3.4.4. Spelling Accent marks

Some languages like in English, while writing some accent marks may appear to abbreviate two words for example in the union of the words do and *not* in an informal writing can be written as *don't*. In contrast to English, in Spanish the grammar rules establish the use of different orthographic accent marks in words according to its structure. So, the spelling accents used in Spanish language are the following: *á*, *é*, *í*, *ó*, *ú*, *ü* and *ñ* (Tellez et al., 2017). The same way that upper- and lower-case combination make the same word look different, it can also happen with the proper usage of the accent marks, in this case as agents write the recommendations in a row, they can skip the grammar rules and for example the word relation in English should be written as *relación* but sometimes appear without the spelling accent mark as *relacion.* In consequence, they can be treated as two different words and this is not desirable because they may produce the same noisy negative effect over the vector space representation of the word if the same word is written with or without spelling accent mark in different documents.

3.4.5. Numbers

Looking forward the purpose of this project the presence of numbers may not give us a clue about the approval or disapproval given by the credit risk agent. So, following the line of removing all what can produce noise, all numbers are removed from text.

3.4.6. Punctuation

Tasks such as sentiment analysis on informal texts like social media can benefit from the use of punctuation. Thus, the usage of exclamation marks probably may indicate that the writer expresses emotion and exaltation of the spirit or if question marks appear, they may indicate doubt and concern.

Regarding the origin of the texts we are analyzing, it is concluded that the presence of orthographic accent marks tends to be in its majority by stops or full stops in the text (.) or commas that indicate

the continuous of an idea. Finally, as they no longer contribute to the classification of the texts, they are removed from the recommendation documents.

Regarding the maximum of 512 tokens length limitation that BERT model has, Table 3 shows how the recommendations are distributed by the length of the text after data preprocessing and setting it to lower case, removing accent marks, numbers and punctuations. The results express that 86,29% of the data base meets the condition of having 512 or less words, but this does not mean that they will all fulfill BERTs limitation. As it uses word piece methodology a word can be separated in several tokens. Even though it represents a huge portion of the database.

Number of words	Count of recommendations	Participation
<=512	117043	86,29%
512 -1000	15390	11,35%
1000 - 2000	3057	2,25%
> 2000	143	0,11%
Total	**135633**	**100,00%**

Table 3: Distribution of recommendations by length of texts

3.5. MODELS

Recent empirical improvements due to transfer learning with language models have demonstrated that rich, unsupervised pre-training is an integral part of many language understanding systems (Devlin et al., 2018). One of the major advantages of using these pre-trained models like the one developed and published in May 2019 by Google AI language team is having a model that can understand general natural language context which.

3.5.1. BERT

Pre-trained BERT model uses a "Masked Language Model" where some random tokens are masked from the input text and the objective is to predict the original vocabulary id of the masked word based only on its context (Devlin et al., 2018). This lets the model understand both left and right content around the world at the same time and avoiding left-to-right and right-to-left separately concatenation models such ELMO. This structure allows training a deep bi-directional Transformer proposed by Google in 2017 which uses self-attention mechanisms. There are available different versions of pre-trained BERT models but there are two general types: BERT-Base which is structured with 12 layers, 768 as hidden size, 12 self-attention heads and 110M of total parameters. At the same time, there is a more extended model which is BERT-Large which was trained with 24 layers, 1024 hidden size, 16 self-attention heads and 340M parameters.

BERT models use WordPiece tokenization proposed in the Neural Machine Translation learning approach published by Google. This tokenization divides words into a limited set of common sub-word units (Wu et al., 2016). Later on, each token is represented as a 768 dimension vector. As said before, for pre-trained BERT models some random tokens are masked and then those tokens are predicted using a deep bi-directional Transformer Encoder. In the last part, the final hidden vectors for those tokens are fed into an output softmax (Devlin et al., 2018). One singularity of BERT refers to that in the

process, types of tokens are placed. The first one is [CLS] which indicates which indicates the start of a new example and [SEP] which refers to the separation token for some special cases like question/answer task.

The pre-trained model used in this project was the Multilingual BERT Base, whose configuration will be used to do fine-tuning for a single text which will represent the recommendation written by the credit analyst. In this case, the token representations are fed into an output layer for token level tasks and the [CLS] representation is fed into an output layer for classification (Devlin et al., 2018) as shown in figure 1.

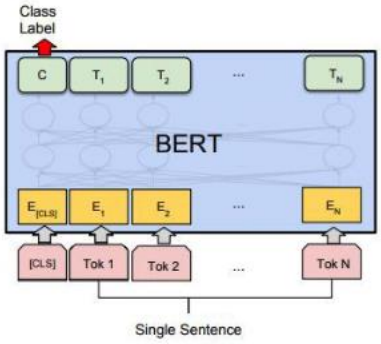

Figure 1. Single sentence BERT classification model

3.5.2. Proposed Model 1

The first approach to solve the text classification problem raised here was to use BERT single sentence classification model (Reimers & Gurevych, 2019)(MacAvaney, Cohan, Yates, & Goharian, 2019). Due to computational restrictions, neither random nor grid search was performed even though every model should consider hyperparameter optimization in order to be tested in different scenarios and to help the machine learning algorithm perform better. BERT single sentence classification model was trained with defined parameters: batch size 32, learning rate of 2e-5 and 3 epochs. In addition, BERT model has a restriction over maximum sequence length where it can be 512 maximum after tokenizing (Devlin et al., 2018; MacAvaney et al., 2019). In this case, stop-words exclusion mitigated the negative impact of restrictions on the execution of the model. However, some texts remained with more than 512 words after cleaning. In response, the model shortens the text when reaches the maximum sequence length. Finally, cross entropy is stablished as loss function for the multi-label classification problem.

3.5.3. Proposed model 2

After developing BERT, the Computer Science Department of the Technische Universität Darmstadt in Germany, developed Sentence-BERT which is a modification of the pretrained BERT network. It uses twin and triplet network structures to derive meaningful sentence embeddings that can be compared using cosine-similarity (Reimers & Gurevych, 2019).

For the second experiment, the sentence-BERT was used. BERT embedding model was fine-tuned with training recommendations data set and a mean pooling layer was activated to get one fixed sized vector for the whole recommendation text. Finally, this document embedding fed a 4 layer fully

connected neural network with RELU activation function and 128 nodes over hidden layers, followed by a softmax output layer. Loss remains as cross entropy with logits function, stochastic gradient descent was chosen as optimizer with an initial learning rate of 0.08. Batch size is set at 32 and 100 epochs tested as running time which was considerably lower than the previous model.

3.6. TECHNOLOGY

Coding was made with python 3.7 and models were trained with a laptop with 8GB of Ram. Some of the basic packages were *Pandas* and *NumPy* for general tasks. In addition, for data transformation *NLTK* stop-words were removed from the recommendation texts. After that, for modeling *BERT* package was used for the first proposed model and *Sentence-Transformer* package in conjunction with *Keras* ran the sequential neural network.

3.7. EVALUATION

Text classification model performance metrics does not differ from traditional classification models. There are some standard metrics that summarize and compare the real label of the text if it is positive, negative or partial and the category predicted by the model. The most popular performance measures come from the confusion matrix where accuracy, precision and recall are calculated (Kowsari et al., 2019).

The accuracy measures the fraction of the correct predictions divided by the total predictions. In unbalanced data sets, this metric should be double checked since a high accuracy does not mean the category of interest has been well predicted. As it is a minority-class the well predicted examples where not the ones of the relevant label. Equation 1 shows how accuracy is calculated based on Table 2 which contains the good and bad classifications made by the algorithm compared with actual results. This table is known as confusion matrix.

		Actual Class	
		Positive	Negative
Predicted Class	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Table 4: Confusion matrix

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Equation 1: Accuracy

Precision measures the fraction of correct predicted examples of the category divided by the total cases predicted in that category. Unlike accuracy, precision measure will only look at the proportion of good classifications divided by the predicted number of cases in that category. Despite a data set is unbalanced it will give the precision of prediction for each category. Equation 2 shows the equation to get precision based on confusion matrix.

$$precision = \frac{\sum_{l=1}^{L} TP_l}{\sum_{l=1}^{L} TP_l + FP_l}$$

Equation 2: Precision

Finally, recall shows the fraction detected of all actual positive cases (a specific category). When misclassifying results in high cost, this metric lets the researcher identify how well you are predicting each class. Equation 3 shows the equation for recall referring to the confusion matrix in Table 2.

$$recall = \frac{\sum_{l=1}^{L} TP_l}{\sum_{l=1}^{L} TP_l + FN_l}$$

Equation 3: Recall

4. RESULTS AND DISCUSSION

The focus of the analytical team in the Credit Risk Department has been in continuous transformation through recent years because of the machine learning algorithms evolution. Even though algorithms like random forest have existed for more than 50 years, the restrictions of computational capacity have been limited because data science field has allowed banks to use these methodologies for more accurate classification of clients. All the Credit Risk Department efforts have bulked to renew scoring rates models and default prediction analysis. In consequence, alternative projects like this one have not been prioritized in the annual agenda; however the results of this experiment will help to show the value of working on non-traditional models and tasks to the managers.

The fear facing new challenges is high, even more when they are proposed in a traditional culture as the financial industry. The first results obtained from the data analysis showed that almost 50% of the recommendations from January 2014 to January 2019 had a fixed structure and a formatted text, which, at first sight, can support important conclusions. The improvement when making the exploratory data analysis was to make evident that almost 50% of the data base had the explicit sentence declaring whether the credit rick agent recommendation was positive, negative or partial. This fact made the classification easier and more accurate. In addition, the direct tagging of almost the half of the data base gave a first insight of the recommendations categories distribution.

Afterwards, the other half of the data base became the object of this project. Tagged examples of 1.181 recommendation texts had a distribution of: 67% for positive, 23.5% corresponded to partial recommendation tag and 9.5% were negative recommendations. As recommendations were tagged in a random way, it is possible to see that the distributions of the two examples differ mostly in the participation on the negative group but have more positive tagged samples. At the end, there was no expectation regarding both class distributions being similar as it is impossible to know the tangible population distribution.

The partition train-test for the model was defined at 70-30 participation, 70% of the tagged data base was used for training and 30% (355 examples) out of the 1.181 examples were used to test the trained model. Confusion matrices are shown below in tables 5 and 6. At the end table 7 will show the aggregated performance metrics for each model.

Real / Predicted	Positive	Partial	Negative
Positive	165	42	0
Neutral	15	69	0
Negative	9	55	0

Table 5: Confusion matrix for single sentence BERT classification model

Analyzing table 5, it can be concluded that single-sentence BERT classification model performed worse than poorly when classifying negative recommendations as it did not classify any of the 355 test

examples in the category. Rather, most of the actual negative recommendations were interpreted as partial recommendation and some few as positive, this shows that the model can understand that the actual negative texts differ more from the positive than the partial recommendations. These results can be the consequence of the small number of negative examples in the tagged data set as they only represent the 9,5% compared to the positive (67%) and partial (23,5%).

Real / Predicted	Positive	Partial	Negative
Positive	170	29	8
Neutral	23	43	18
Negative	13	22	29

Table 6: Confusion matrix for sentence-BERT fully connected neural network

Table 6 shows the confusion matrix result of the Sentence-BERT fully connected neural network. Without doubt, at first sight it is possible to say that it performed better than the single-sentence BERT classification model, as it was able to classify texts into all three categories and the majority of the examples. This means the diagonal of the confusion matrix accumulates the majority of total cases. Even though this model did classify some negative examples correctly, it can be seen that the model did not even predict well half of the examples in this category.

	Single sentence BERT classification model	Sentence-BERT fully connected neural network
Accuracy	65,9%	68,16%
Precision - Positive class	87,3%	82,52%
Precision - Partial class	41,5%	45,74%
Precision - Negative class	0%	52,72%
Recall - Positive class	79,71%	82,12%
Recall – Partial class	82,14%	51,19%
Recall – Negative class	0%	45,31%

Table 7 : Model performance metrics

Now looking at both models aggregated performance metrics in table 7, It is amazing that in terms of accuracy the difference is minimal. The thing is that accuracy is not a good evaluation metric when having an unbalanced dataset because the algorithm predicts well the category when it has the

majority of examples so, it will weigh more than the good predictions of the small class. This is why precision and recall of each category are more exact and decisive when evaluating a multi-label classification model.

The precision metric in a category will show the percentage of good predicted examples of that class divided by the total examples predicted in that category. For partial and negative classes, the sentence-BERT fully connected neural network performed better than the first model, this result is very important as they are the minority classes and will be explained later. Moreover, there is the recall metric which presents the percentage of good predictions divided by the real number of examples in that class. Regarding recall it is shown that for partial class the single-sentence BERT classification model behaves better with 82,14% of desirable recall, but unfortunately for negative class this model, it has 0% recall as it did not classify any of the examples in that group.

Having the complete vision of the performance of both models, it is important to say that the sentence-BERT fully connected neural network is the model that behaved better. First, it was able to identify particular characteristics and special patterns for each category. This is very powerful because its level of discrimination between classes is higher and the perfect classification model is able to discriminate one group from another, contrary to what happened in the single-sentence BERT that misclassifies big percentage of negative texts as partial. Thus, it can be interpreted as a confusion of the algorithm between the characteristics of a negative and a partial recommendation. Second, the sentence-BERT model behaves better than the first model when regarding the business analysis, comparing all the three steps of the credit application process and adding the client credit behavior after the disbursement of the credit product. An undesirable case will be the one where the credit risk agent did not recommend the operation and the commercial agent skipped the given recommendation and then the client entered in default and did not pay the loan. For that, it is very imperative to identify a negative recommendation.

Loss value for single sentence BERT classification model was 0.29 for training and 0.27 for test set, much better results than sentence-BERT model that showed 0.44 in training and 0.42 in test. However, single sentence BERT classification model did not classify any recommendation text as negative and positive class was the highest precision, as expected.

Sentence-BERT fully connected neural network spend less time running and accuracy is around 68,17% compared to single sentence classification model that reached 65% of accuracy for test set.

Analyzing predicted and actual values for test set, the model with best results is the sentence-BERT fully connected neural network for classification since it shows a better result in accuracy and precision of the non-majority classes (partial and negative). Nevertheless, for partial class the recall is a better way in the single sentence BERT model, which detected 82,14% of actual partial recommendation class texts.

5. CONCLUSION

This project presents the approach for the construction of a text classification model of recommendation text for SME credit applications and to give an analytical solution to the credit risk area.

The solution of classifying all the recommendation texts into one of three categories must be developed in two steps. The first one is to classify all the text via regular expression which was shown in the initial data analysis. Then the remained texts will be classified by the sentence-BERT neural network classification model. For this second group, it is expected to have an accuracy of 68,16% so that will give a rate of misclassification with 31,84% of the 50% of the database. However, it will represent the 15,92% of the whole database which will represent a high positive impact on the business. It is expected to have an error rate in model developed but, in this case, the business will change from having no idea of what the behavior is to a very good first version of the classification model. Here, it is important to highlight that the distribution of classes affected the results of the algorithm. To solve this issue, more tagged examples were useful for the model to understand the conditions and the characteristics of each category; and as a consequence, it discriminated better one category from the other.

The model that best performs was sentence-BERT which got word embeddings through BERT and a pooling layer was added to capture semantically similar features into one and to get a fixed dimension vector for all the text. Limited computational resources were the big constraint when developing the project.

In order to improve the performance of the classification model, it is important that in future versions of this model an exhaustive data cleaning helps to remove noisy words that do not contribute to the classification model. Also, hyperparameter tuning should be tested and compared to the current results. Finally, it is relevant to increase the number of tagged samples; this will help the algorithm understand better the characteristics of a positive, partial and negative recommendation text.

6. LIMITATIONS AND RECOMMENDATION FOR FUTURE WORKS

Free cloud computing tools like Google Colab and Azure Notebooks gives access to any researcher to use the computational capacity of GPUs and TPUs to train algorithms and get faster results rather than use their default CPU. Even though these tools offer more computational capacity, the credit application recommendation texts are considered confidential information, so it was forbidden to upload the data base to Microsoft or google cloud to train the model there even if it was classified in the cloud as private project. As a result, this constitutes the biggest constraint in the project as training models took very long.

A consequence for the limitation on computational capacity is to run the models. Random hyperparameter search was explored but laptop memory was not able to finish the process. This task is very important in the development of a ML problem because it gives the researcher a range of space where the parameters behave well and give better results. In further works, this should be done to explore more than the two options presented in this project.

Since both models where based on BERTs structure, one of the constraints when implementing BERT is the maximum sequence length limited to 512 tokens. Despite the effort to remove noisy information from the recommendation texts by removing numbers, special characters and stop-words, some texts exceeded the limit. For further versions it is recommended to work deeper with text cleaning to avoid non relevant words get into the algorithm.

Last of all, having more examples to train and test the model generally results in a better-performed model. This is because the algorithm will detect more easily the patterns and conditions of the texts that correspond to one class or another. Also, as this is the first work that has been made over this data it its unknown the natural distribution among classes, so having more tagged examples will give a first insight of the behavior of the credit recommendation cycle, for example supporting strategical expansions or contractions about the credit market on the analyzed window.

7. BIBLIOGRAPHY

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., … Asari, V. K. (2019). A State-of-the-Art Survey on Deep Learning Theory and Architectures. Electronics, 8(3), 292. https://doi.org/10.3390/electronics8030292

Altınel, B., & Ganiz, M. C. (2018). Semantic text classification: A survey of past and recent advances. Information Processing and Management, 54(6), 1129–1153. https://doi.org/10.1016/j.ipm.2018.08.001

Ay Karakuş, B., Talo, M., Hallaç, İ. R., & Aydin, G. (2018). Evaluating deep learning models for sentiment classification. Concurrency Computation , 30(21), 1–14. https://doi.org/10.1002/cpe.4783

Cambria, E., Poria, S., Gelbukh, A., & Thelwall, M. (2017). Sentiment Analysis Is a Big Suitcase. IEEE Intelligent Systems, 32(6), 74–80. https://doi.org/10.1109/MIS.2017.4531228

Conneau, A. (2017). Very Deep Convolutional Networks for Text Classification. 1(2001), 1107–1116.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (Mlm). Retrieved from http://arxiv.org/abs/1810.04805

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2019). Learning word vectors for 157 languages. LREC 2018 - 11th International Conference on Language Resources and Evaluation, 3483–3487.

Gupta, S., & Gupta, S. K. (2019). Abstractive summarization: An overview of the state of the art. Expert Systems with Applications, 121, 49–65. https://doi.org/10.1016/j.eswa.2018.12.011

Hashimi, H., Hafez, A., & Mathkour, H. (2015). Computers in Human Behavior Selection criteria for text mining approaches. 51, 729–733. https://doi.org/10.1016/j.chb.2014.10.062

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), 1, 328–339. https://doi.org/10.18653/v1/p18-1031

Jianqiang, Z., & Xiaolin, G. (2017). Comparison research on text pre-processing methods on twitter sentiment analysis. IEEE Access, 5, 2870–2879. https://doi.org/10.1109/ACCESS.2017.2672677

Kim, Y. (2014). Convolutional neural networks for sentence classification. EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 1746–1751. https://doi.org/10.3115/v1/d14-1181

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L. E., & Brown, D. E. (2019). Text Classification Algorithms: A Survey. 1–68. https://doi.org/10.3390/info10040150

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent convolutional neural networks for text classification. Proceedings of the National Conference on Artificial Intelligence, 3, 2267–2273.

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. https://doi.org/10.1038/nature14539

MacAvaney, S., Cohan, A., Yates, A., & Goharian, N. (2019). CEDR: Contextualized embeddings for document ranking. SIGIR 2019 - Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 1101–1104. https://doi.org/10.1145/3331184.3331317

Mäntylä, M. V., Graziotin, D., & Kuutila, M. (2018). The evolution of sentiment analysis—A review of research topics, venues, and top cited papers. Computer Science Review, 27, 16–32. https://doi.org/10.1016/j.cosrev.2017.10.002

Mikolov, Tomas ; Le, Q. (2014). Distributed Representations of Sentences and Documents. 32, 1–5. https://doi.org/10.1145/2740908.2742760

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. 1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings, 1–12.

Mohammad, F. (2018). Is preprocessing of text really worth your time for toxic comment classification? 2018 World Congress in Computer Science, Computer Engineering and Applied Computing, CSCE 2018 - Proceedings of the 2018 International Conference on Artificial Intelligence, ICAI 2018, 447–453.

Otter, D. W., Medina, J. R., & Kalita, J. K. (2018). A Survey of the Usages of Deep Learning in Natural Language Processing. Retrieved from http://arxiv.org/abs/1807.10854

Pakhchanyan, S. (2016). Operational Risk Management in Financial Institutions : A Literature Review. https://doi.org/10.3390/ijfs4040020

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. 2227–2237. https://doi.org/10.18653/v1/n18-1202

Pons, E., Braun, L. M. M., Hunink, M. G. M., & Kors, J. A. (2016). Natural language processing in radiology: A systematic review. Radiology, 279(2), 329–343. https://doi.org/10.1148/radiol.16142770

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Retrieved from http://arxiv.org/abs/1908.10084

Saif, H., Fernandez, M., He, Y., & Alani, H. (2014). On stopwords, filtering and data sparsity for sentiment analysis of twitter. Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014, (i), 810–817.

Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. Neural Networks, Vol. 61,

pp. 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Silva, C., & Ribeiro, B. (2003). The Importance of Stop Word Removal on Recall Values in Text Categorization. Proceedings of the International Joint Conference on Neural Networks, 3, 1661–1666.

Tellez, E. S., Miranda-Jiménez, S., Graff, M., Moctezuma, D., Siordia, O. S., & Villaseñor, E. A. (2017). A case study of Spanish text transformations for twitter sentiment analysis. Expert Systems with Applications, 81, 457–471. https://doi.org/10.1016/j.eswa.2017.03.071

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 2017-Decem(Nips), 5999–6009.

Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., … Liu, H. (2018). A comparison of word embeddings for the biomedical natural language processing. Journal of Biomedical Informatics, 87(April), 12–20. https://doi.org/10.1016/j.jbi.2018.09.008

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. In Journal of Big Data. https://doi.org/10.1186/s40537-016-0043-6

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., … Dean, J. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. 1–23. Retrieved from http://arxiv.org/abs/1609.08144

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing [Review Article]. IEEE Computational Intelligence Magazine, 13(3), 55–75. https://doi.org/10.1109/MCI.2018.2840738

Zhou, P., & Qi, Z. (2016). Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling. 2(1), 3485–3495.