









# Using Transfer Learning To Classify Long Unstructured Texts with Small Amounts of Labeled Data

Carlos Alberto Alvares Rocha<sup>\*,1,8</sup><sup>a</sup>, Marcos Vinícius Pinheiro Dib<sup>1,2</sup><sup>b</sup>, Li Weigang<sup>\*,1,2</sup><sup>c</sup>,  
Andrea Ferreira Portela Nunes<sup>1,3</sup><sup>d</sup>, Allan Victor Almeida Faria<sup>1,4</sup><sup>e</sup>, Daniel Oliveira Cajueiro<sup>1,5</sup><sup>f</sup>,  
Maísa Kely de Melo<sup>1,6</sup><sup>g</sup> and Victor Rafael Rezende Celestino<sup>1,7</sup><sup>h</sup>

<sup>1</sup>LAMFO - Lab. of ML in Finance and Organizations, University of Brasilia, Campus Darcy Ribeiro, Brasilia, Brazil

<sup>2</sup>TransLab, Department of Computer Science, University of Brasilia, Campus Darcy Ribeiro, Brasilia, Brazil

<sup>3</sup>Ministry of Science, Technology and Innovation of Brazil, Federal District, Brazil

<sup>4</sup>Department of Statistics, University of Brasília, Federal District, Brazil

<sup>5</sup>Department of Economics, University of Brasilia, Federal District, Brazil

<sup>6</sup>Department of Mathematics, Instituto Federal de Minas Gerais Campus Formiga, Formiga, Brazil

<sup>7</sup>Department of Business Administration, University of Brasilia, Federal District, Brazil

<sup>8</sup>PPMEC, Faculty of Technology, University of Brasilia, Federal District, Brazil


**Keywords:** CNN, Deep Learning, MCTI, Longformer, Web Long-text Classification, LSTM, Transfer-learning, Word2vec.


**Abstract:** Text classification is a traditional problem in Natural Language Processing (NLP). Most of the state-of-the-art implementations require high-quality, voluminous, labeled data. Pre-trained models on large corpora have shown beneficial for text classification and other NLP tasks, but they can only take a limited amount of symbols as input. This is a real case study that explores different machine learning strategies to classify a small amount of long, unstructured, and uneven data to find a proper method with good performance. The collected data includes texts of financing opportunities the international R&D funding organizations provided on their websites. The main goal is to find international R&D funding eligible for Brazilian researchers, sponsored by the Ministry of Science, Technology and Innovation. We use pre-training and word embedding solutions to learn the relationship of the words from other datasets with considerable similarity and larger scale. Then, using the acquired features, based on the available dataset from MCTI, we apply transfer learning plus deep learning models to improve the comprehension of each sentence. Compared to the baseline accuracy rate of 81%, based on the available datasets, and the 85% accuracy rate achieved through a Transformer-based approach, the Word2Vec-based approach improved the accuracy rate to 88%. The research results serve as a successful case of artificial intelligence in a federal government application.


## 1 INTRODUCTION


In Natural Language Processing (NLP), a traditional problem is text classification. It consists in predicting/assigning a predefined category(s) to an input text.


For this goal, a crucial intermediate task is text representation. Literature covers various neural models for learning text representation, from convolutional (Zhang et al., 2017; Shen et al., 2018) and recurrent models (Yogatama et al., 2017; Seo et al., 2017) to attention mechanisms (Yang et al., 2016; Lin et al., 2017). Alternatively, pre-trained models on large corpora have shown beneficial for text classification and other NLP tasks, possibly preventing the need to train a new model from scratch. One type of pre-trained model is word embeddings, such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). Another option is contextualized word embeddings, such as CoVe (McCann et al., 2017) and ELMO


<sup>a</sup>  <https://orcid.org/0000-0003-0481-6907>


<sup>b</sup>  <https://orcid.org/0000-0003-0998-0597>


<sup>c</sup>  <https://orcid.org/0000-0003-1826-1850>

<sup>d</sup>  <https://orcid.org/0000-0002-3668-8535>

<sup>e</sup>  <https://orcid.org/0000-0002-4300-9334>

<sup>f</sup>  <https://orcid.org/0000-0001-5898-1655>

<sup>g</sup>  <https://orcid.org/0000-0001-8120-9778>

<sup>h</sup>  <https://orcid.org/0000-0001-5913-2997>

(Peters et al., 2018).

These word embeddings contextualized or not, often function as additional features to aid the main task. More recent studies have shown pre-trained language models to be effective in learning common language representations by utilizing a large amount of unlabeled data, such as Generative Pre-trained Transformer - GPT (Radford et al., 2018; Brown et al., 2020) and Bidirectional Encoder Representations from Transformers - BERT (Devlin et al., 2018).

Most of these models go through the training process using a finite set of data, and exceptional results depend on a volume of high-quality, labeled data. The possible variations in quantity or quality of the data can influence the results. In real applications, these models can yield unexpected and unsatisfactory results, affecting the robustness of the solution (Gron, 2017).

The data used to train these models can be unstructured text, the most widely available information on the internet. As much as they are easy to comprehend by humans, they are a challenging input for machines. Therefore, there is a need to develop algorithms and methods capable of processing large amounts of text for various applications (Allahyari et al., 2017). The data can be obtained through information extraction techniques and data mining (Han and Kamber, 2000). Data extraction from diverse platforms results in a nonuniform dataset with unstructured data, where it is not clear where the most relevant information is.

Although Transformer-based approaches have achieved state-of-the-art results in NLP tasks, they are well-suited to deal with relatively short sequences. These models typically limit inputs to  $n = 512$  tokens due to the  $O(n^2)$  cost of attention hindering their ability to classify long texts (Ainslie et al., 2020). However, there are many NLP applications built around large blocks of text. An example is topic identification of spoken conversations (Hazen, 2011; Kesiraju et al., 2016; Pappagari et al., 2018) and customer satisfaction prediction of call centers (Chowdhury et al., 2016; Luque et al., 2017; Park and Gates, 2009; Meinzer et al., 2017). In the case of call center conversations, the input can vary from short chats to longer ones involving agents trying to solve complex issues that the customers experience, resulting in calls taking as long as an hour or more. An automatic speech recognition (ASR) system transcribes these calls. Such a transcript can exceed the length of 5000 words. Thus ETC (Ainslie et al., 2020) and Longformer (Beltagy et al., 2020) were proposed and obtained state-of-the-art results balancing performance and memory usage.

Another trend employed in NLP tasks is the use of Transfer Learning techniques that allows using the knowledge of an original domain and provides a means of transferring said knowledge to the destination domain to increase the data coverage. While more commonly found in image processing applications, it was shown to be efficient in NLP applications by allowing the sharing of knowledge like similarity in linguistic representation (Ruder et al., 2019). Transfer learning takes the knowledge previously acquired in the original domain and continues the training with new data.

This paper will focus on a more specific problem, creating a Research Financing Products Portfolio (FPP) outside of the Union budget, supported by the Brazilian Ministry of Science, Technology, and Innovation (MCTI). The problem description and conceptual model of FPP/MCTI are shown in Figure 1. The input data includes the text of financing opportunities offered by many institutions worldwide on their websites, such as scholarships, grants, fellowships, and others. A small part of these data was manually labeled by the ministry staff and used in the supervision and training of the classification model, which achieves the accuracy goal defined by the discrimination of the opportunities that allows research financing for Brazilian projects. Due to the nature of the data collection, it has the following characteristics:

- Most of the data is unstructured and nonuniform.
- Texts with high variance in length, reaching up to 5000 tokens/words.
- A short amount of labeled data.

In this article, we explore different Artificial Intelligence (AI) strategies to classify a small amount of long, unstructured, and uneven data to find the appropriate method with good performance.

As the main contribution of this research, we use pre-training and word embedding solutions to learn the relationship of the words from other datasets with considerable similarity and larger scale. Then, using the acquired features, based on the available dataset from MCTI, we apply transfer learning plus deep learning models to improve the comprehension of each sentence to describe the information of the websites. Compared to the baseline accuracy rate of 86%, based on the available datasets, the proposed Word2Vec-based approach in the classification improves the accuracy rate to 88%. All training and testing were done by using Google Colab Pro.

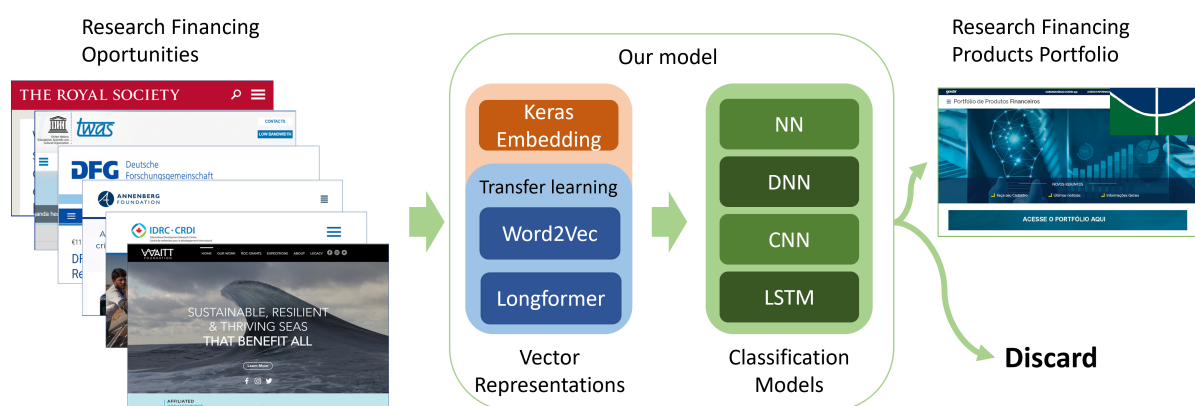


Figure 1: FPP/MCTI classification model.

## 2 PROBLEM DESCRIPTION

Due to recurrent budget cuts in recent years, the MCTI has been looking for ways to develop solutions to finance and promote Science and Technology projects outside the Union budget. A recent partnership was made between MCTI and the University of Brasilia (UnB) to research a semi-automatic system based on AI and data science to assist scholars in finding funding sources for technical projects.

The Research Financing Products Portfolio (FPP) is a system maintained by the Secretariat of Financial Structures and Projects (SEFIP) of MCTI, whose objective is to promote scientific research and raise resources and financing. The FPP presents information on opportunities offered by institutions, foundations, and banks, such as Development Agencies and Multilateral Banks. However, the tools for searching and updating system information are currently manual, making the process long and prone to errors. The expectation is that the system's modernization through Machine Learning (ML) and NLP solutions will provide tools for searching and updating data in an automatic and optimized way.

The research aims to determine and develop intelligent and optimized tools to search, process, organize and present data to compose the FPP. It should also implement a solution with user recommendation and interaction so that the experience allows the continuous improvement of the resources' usability using reinforcement learning.

### 2.1 Main Steps of the Project

The implementation of the project is described in the following steps: Data Scraping, Classification, Summarization, and Recommendation.

**Data Scraping:** The first step in automating identifying and evaluating these sources is reading or capturing the data. For this, a technique called data scraping will be used, which will consist of automated access to leading online platforms that provide financing resources for research projects. The idea is to be able to scan data from these websites and collect information regarding opportunities so that they can be used in the subsequent stages of the project.

**Classification:** With data adequately collected and organized, we can use machine learning techniques to classify opportunities to select those open to Brazilian projects. Since the classification step uses data in text format, it will be necessary to use natural language processing techniques so that it is possible to extract contextual information and allow text interpretation by the computer.

**Summarization:** The next step is data summarization, which uses machine learning and natural language processing techniques to summarize texts of selected and classified opportunities. The summarized texts will facilitate the subsequent dissemination of opportunities.

**Reinforcement Learning and Recommendation:** In order to allow the continuous improvement of the proposed solution, the last stage of the project involves the development of an initial prototype of a recommendation system that uses machine learning. The implementation of the recommendation engine will use reinforcement learning, AI techniques, and feedback regarding the users' experience.

### 2.2 Challenges in the Development of the Project

Artificial intelligence, specifically machine learning and deep learning, has come a long way in the last few

decades (Parloff, 2016). The main reason for this evolution is improved computer performance that made previously impossible tasks possible. Another significant factor was the availability on the internet of large masses of data to train specific-purpose models with an ever-increasing number of parameters.

In the context of this project, we can understand that most of the difficulties associated with the solution are related to these technologies' short time of existence. The work will be developed based on disruptive discoveries and results presented in the last five years and can be currently stated as state-of-the-art. When working with such recent research, it is possible to find problems with replication and implementation with different types of data and different architectures.

The first challenge is related to data acquisition in the scraping stage. It involves scanning several online platforms with different structures, so specific scraping codes are required for each platform. A structural update to a platform can make the scraping code obsolete and deliver incorrect or incomplete data.

Another foreseen challenge concerns the quality of the data obtained through scraping. Due to differences in the text's format, language, and writing, from each scraped platform, it will be difficult to guarantee a standardization of the input data for the training steps in machine learning, which can cause a drop in performance or a worsening of results.

The third challenge is obtaining extra-contextual information currently used by civil servants to identify opportunities. Because of its specificity, this information is difficult to share. The historical context of the platform, and different terminology used by the platform, are good examples of this.

In this paper, we focus on developing Machine Learning solutions with a pre-training strategy for the classification problem.

### 2.3 Input Data

Our work begins with the data obtained from scraping techniques, and it is vital to show the format and current state of the initial inputted data. The data used was from over 30 different platforms e.g. The Royal Society, Annenberg foundation, and contained 357 rows, but only 260 of them were labeled as shown in Figure 2. Of the data gathered, we were only interested in the main text content and did not use information related to the website URL, title of the page, and other metadata. The content text averages 800 tokens in length, but it has a high variance, reaching up to 5000 tokens as shown in Figure 3.

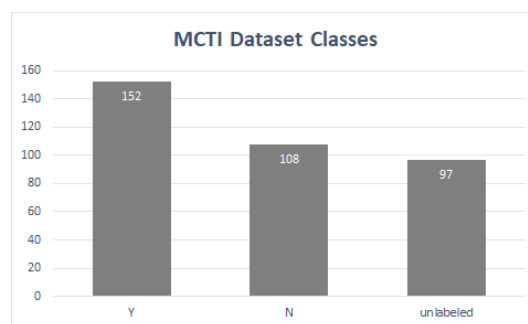


Figure 2: MCTI dataset classes.

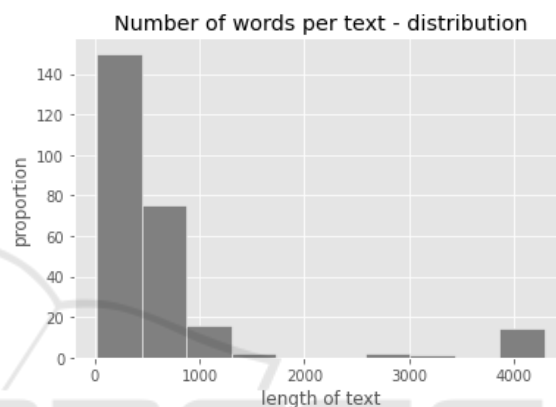


Figure 3: MCTI text length distribution.

## 3 RELATED WORK

In this section, we overview other works related to the classification task in NLP, presenting state-of-the-art works in classification, approaches to long texts, small amounts of labeled data, and transfer-learning.

**Deep Learning Models.** Recent advances in computational power and availability of data granted Deep Learning (DL) a new resurgence (Thompson et al., 2020), more specifically, the progress of artificial Neural Networks (NN), such as Convolutional Neural Networks (CNN) and Long short-term memory (LSTM). These deep neural network models have been used successfully in text and document classification tasks (Minaee et al., 2021).

A DNN is a Deep Neural Network with more hidden layers. These multiple layers of abstraction seem likely to give deep networks a compelling advantage in learning to solve complex pattern recognition problems (Nielsen, 2015). DNNs are composed of connected layers where each layer receives connections from the previous layer and provides connections to the following (Kowsari et al., 2017). The implementa-

tion uses a standard backpropagation algorithm. The output layer has one node for each class to be classified, only one for binary classification, and it is a softmax function. The input needs to be vectorized text using techniques such as word embedding, and the purpose of the network is to learn the relationship between inputs and target spaces using hidden layers (Kowsari et al., 2019).

LSTM stands for Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) and is a neural network based on Recurrent Neural Networks (RNN) with the ability to handle the preservation of short-term memories, specifically circumventing the problem of the vanishing gradient (Pascanu et al., 2013) present in the RNN. It is made of units composed of a cell, an input gate, an output gate, and a forget gate (Gers et al., 2000). Like the RNN, the LSTM contains a chain architecture that considers information from previous nodes, with the difference that it also contains several ports that regulate the amount of information allowed in each node state. This type of architecture associated with word-embeddings has attained excellent results in text classification (Wang et al., 2018; Xiao et al., 2018).

A CNN is a type of Deep Neural Network that applies convolution operations to the input data before feeding it to the NN section. A basic CNN consists of a Convolutional layer, a Pooling layer, and a Fully-connected (FC) or Dense layer. The convolutional layer is responsible for massive data processing using filters and resource maps. The pooling layer is responsible for grouping data to reduce computational complexity and reduce the number of network outputs so that essential characteristics are preserved. The fully connected layer performs the classification task based on features extracted from previous layers (IBM Cloud Education, 2020). While it is most commonly known for image processing (2D) and computer vision (2D to 3D) applications, 1-dimensional convolution exists and has achieved great results in text classification (Kim, 2014).

**Transfer Learning.** The usage of transfer learning in NLP tasks is not new (Pan et al., 2011; Do and Ng, 2005) and has achieved excellent results over the years. The main idea is to transfer knowledge from different source domains to a target domain. A common approach for this is using word vector representations (Ruder et al., 2019; Raina et al., 2007) such as word-embeddings. It is best used when sufficient training data is only available in another domain of interest. In this case, the knowledge transfer could significantly improve learning performance by avoiding expensive data-labeling efforts (Pan and Yang, 2010).

**Pre-training and Fine-tuning.** Within the NLP realm, it has become common to approach problems through transfer learning instead of building a model from scratch by using a model pre-trained on another task and fine-tuning it via further training in the dataset of interest (Mohammed and Ali, 2021; Church et al., 2021). In order to classify texts available on an overload of datasets, it was proposed to use Pre-trained language models (PLM). PLMs are previously trained on a large corpus of text and have some ability to understand text context, such as Transformer-based models (Vaswani et al., 2017).

BERT (Devlin et al., 2018) is still presented as state-of-the-art for most NLP tasks (van den Bulk et al., 2022). Its training is done by conditioning both left and right contexts, simultaneously optimizing for tasks of a masked word and next sentence prediction. BERT-base has an encoder with twelve transformer blocks, twelve self-attention heads, a hidden size of 768, and a maximum input sequence of 512 tokens. In order to perform the classification task, a classification head is included with a simple softmax classifier to return labels' probabilities (Sun et al., 2019). However, this neural network usually only performs when broad information is available in the dataset (Kontonatsios et al., 2020; van Dinter et al., 2021).

In order to process long documents, BERT truncates the text into the max input size (1024 for BERT-large). Another approach presented by (Sun et al., 2019) uses chunking and text fractions to use BERT for long texts. The Longformer (Beltagy et al., 2020) approach uses an attention pattern that combines local and global information while also scaling linearly with the sequence length. It can perform a wide range of document-level NLP tasks without chunking/shortening the long input and without complex architecture to combine information across these chunks achieving state-of-the-art results on the character-level language modeling task. Similarly, ETC (Ainslie et al., 2020) also uses an attention mechanism but differs from the Longformer by combining global-local attention with relative position encodings and flexible masking, enabling it to encode structured inputs similarly as graph neural networks do. Although these implementations performed well when using transformers, they relied on a large number of training data to achieve their results.

Since large-scale machine learning methods or tools such as BERT requires high-performance computing and extremely large-scale data, the tasks faced by our project do not have these resources. Therefore, we must form a technical procedure suitable for this purpose. Based on the existing dataset D1 by MCTI, we find a dataset D2 that has a certain similarity with

D1 and is relatively large in scale. Use D1+D2 data to carry out Pre-training, obtain the corresponding features through Word Embedding, and then fine-tune the DNN models under the guidance of the Transfer learning strategy to achieve Few-shot learning.

**Few-shot Learning.** It is interesting to mention the evolution of the few-example learning. In 1998, the “Once learning” mechanism was proposed for image clustering by one example to simulate human learning behavior (Weigang, 1998) using Self-Organization Map (SOM). This paradigm was also applied to identify the Radar images (Weigang and da Silva, 1999). The researchers (Miller et al., 2000) defined a process to learn from one example through shared densities on transforms. This method used “prior knowledge” to develop a classifier based on only a single training example for each class. Li FeiFei and others (Fei-Fei et al., 2003) developed “One-shot learning” to use knowledge about object categories to classify new objects as humans do. After then, the concept was generalized as “Few-shot learning,” accepted by the community, and applied successfully in NLP applications (Brown et al., 2020).

## 4 USING TRADITIONAL DEEP LEARNING METHODS

There have been previous attempts to classify the financing product dataset using non-machine learning approaches. One solution using Naive Bayes and TF-IDF achieved an accuracy of 86% (Silva et al., 2021). Another solution employed keyword search, in which the presence of selected keywords such as Brazil, South America, and others would determine whether it was an eligible opportunity. This solution achieved an accuracy of about 78%. We begin this work by establishing a traditional machine learning model and normalizing the input data, using word embeddings and classification models already consolidated in the NLP community.

### 4.1 Data Normalization

*Normalization* is a preprocessing stage often applied in machine learning systems. The process consists of scaling the data to a needed interval. Also called data scaling, normalization scales the data to study little insights and valuable relationships and to work with the best features of the data (Sree and Bindu, 2018). While not every dataset requires data normalization, when features have different ranges of val-

ues, the model might be unable to learn or oscillate back and forth for a long time before finally finding its way to the global/local minimum. Different features with similar ranges of values allow gradient descents to converge more quickly.

Due to the provided data being composed of unstructured and nonuniform texts, normalizing the data became pivotal for defining the development baseline. Normalized data can incorporate more easily into the other layers of the application. The normalization applied was Text Normalization, such as removing HTML special characters and capital letters.

## 4.2 NLP Classification Models

### 4.2.1 Word Embeddings (WE)

The first layer of the proposed model is an embedding layer. Word embedding is a method of extracting features from the data, which can replace one-hot encoding with dimensional reduction. While dealing with textual data, words need to be converted into numbers before being fed into a machine learning model. A simple way is to use one-hot encoding to convert categorical features into numbers. For example, one-hot encoding would transform the input integer values of 0, 1, and 2 to the vectors [1, 0, 0], [0, 1, 0], and [0, 0, 1] respectively (Heaton, 2020). This method results in a dummy feature for each word, which means, for example, 10,000 features for a vocabulary of 10,000 words. This approach is not feasible as it demands large storage space for the word vectors and reduces model efficiency. The embedding layer lets us convert each word into a fixed length vector of defined size. The resultant vector is a dense one with real values instead of just 0's and 1's. In that way, the embedded layer is like a look-up table. The words are the keys in this table, while the dense word vectors are the values.

### 4.2.2 Models

After the embedding, which is just essentially data preprocessing, it is necessary to develop the project further to analyze the input text and classify whether it is a valid research funding opportunity for Brazilian or not. A neural network architecture can be appended to the embedding layer. Various architectures have different performances and training times. For the project, the best option would be chosen empirically upon comparing the results of 4 distinct architectures: Neural Network (NN), Deep Neural Network (DNN), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN). Figure 4 shows the structure of the models.

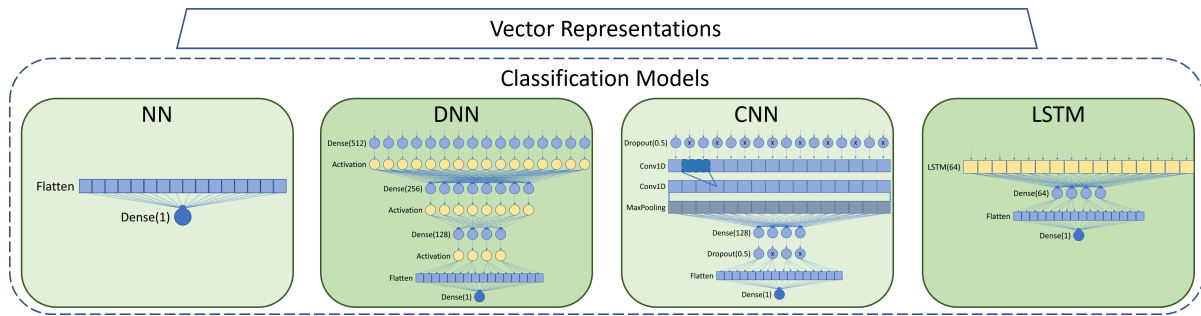


Figure 4: Classification models.

A neural network (NN) here is a simple feedforward neural network with only a single hidden layer, usually called "shallow." Shallow NNs are often limited in the complexity of the problems they can be trained to solve well. Our NN model comprises a simple Flatten layer and a dense classification layer. Our DNN model is composed of a sequence of dense and activation layers varying in size from 512 to 256, to finally 128, and then a Flatten layer feeding into the final dense classification layer. Our CNN model uses a dropout layer feeding into a couple of Conv1D layers and then a MaxPooling layer. After that, we use a hidden layer composed of a dense layer of size 128, followed by another dropout layer, and finally, the Flatten and final dense classification layer. Our LSTM model is formed by an LSTM layer with 64 cells connected to a hidden layer composed of a dense layer of size 64 and, finally, the Flatten and final dense classification layer.

#### 4.2.3 Results from WE + DNN Models

The word embeddings used were single layers of 64 dimensions trained alongside the complete system. We tested the network with architectures of NN, DNN, CNN, and LSTM, and the results with related metrics such as accuracy, precision, recall, and F1 Score are listed in Table 1. When examining the table, it is possible to verify that all the models achieved accuracies already superior to the 78% baseline, reaching values close to 85% in the CNN architecture. Another interesting point to analyze is that this model obtained a precision of 100%, meaning that all opportunities identified as eligible were correctly identified.

Table 1: Results from WE + ML models.

ML Model	Accuracy	F1 Score	Precision	Recall
NN	0.8269	0.8620	0.8212	0.9129
DNN	0.8269	0.8650	0.8952	0.8447
CNN	0.8462	0.8756	1.0000	0.7803
LSTM	0.8269	0.8675	0.8276	0.9129

## 5 USING PRE-TRAINING AND TRANSFER-LEARNING

With the motivation to increase accuracy obtained with baseline implementation, we implemented a transfer learning strategy under the assumption that small data available for training was insufficient for adequate embedding training. In this context, we considered two approaches: i) pre-training word-embeddings using similar datasets for text classification; ii) using transformers and attention mechanisms (Longformer) to create contextualized embeddings. These pre-training approaches can be visualized in Figure 5.

### 5.1 Pre-training Word2Vec Embeddings

Unsupervised training of word-embeddings using the word2vec technique has shown to be very successful in classification tasks (Zhang et al., 2015; Lilleberg et al., 2015).

We need to train word embeddings that can represent the context of the words in our dataset. Since labeled data is scarce, we trained word-embeddings in an unsupervised manner using other datasets that contain most of the words it needs to learn.

#### 5.1.1 Pre-training Datasets

The idea implemented was based on introducing better and better-trained word embeddings in the model. We need larger datasets that contain the words we want to have in vectorial representation in their corpus. For an additional dataset to be applied to improve word-embedding training, it must be compatible with the dataset used to train the classifier. A completely unrelated set of sentences would add, at best, a few examples of the words present in MCTI's different contexts. To evaluate the possible extra datasets, we propose a simple formula:

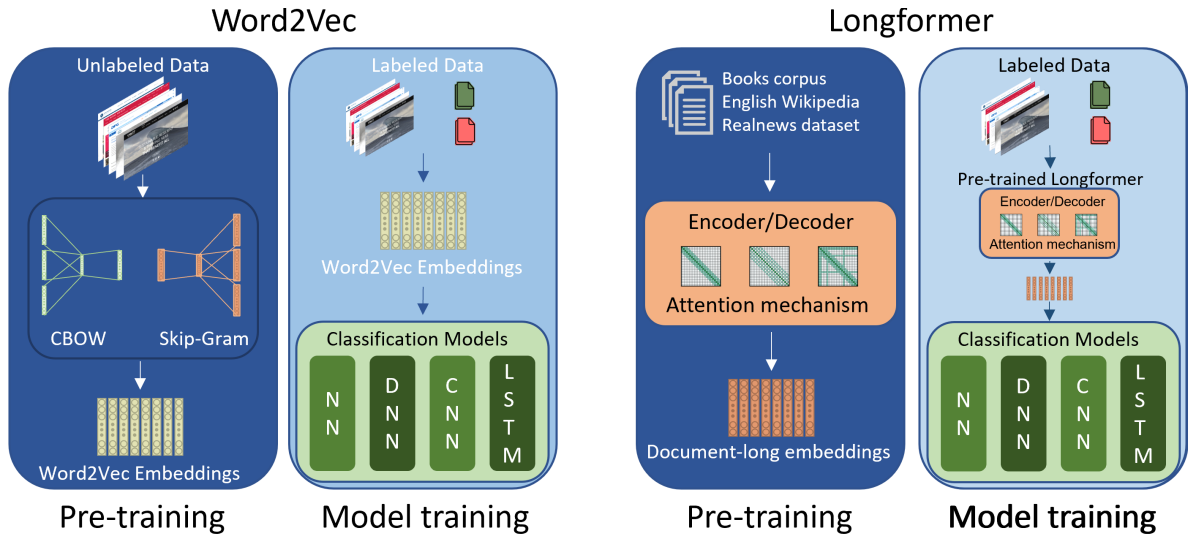


Figure 5: Pre-training models.

$$P_C = \frac{\#\{\{Base\} \cap \{New\}\}}{\#\{Base\}} * 100 \quad (1)$$

where  $\{Base\}$  is the set (unique tokens) of the MCTI dataset,  $\{New\}$  is the set of the target additional dataset, and  $\#$  denotes the number of elements in a Set. This formula calculates the percentage of the words in the original dataset present in the new dataset. The maximum number is 100 (all tokens are present in the new dataset).

The equation (1) yields a percentage of similarity that will help assess which datasets to use for the pre-training. This determination is essential once we do not have the computational power to pre-train with massive data like professional solutions such as BERT and GPT (Weigang et al., 2022).

We searched for datasets from the Kaggle, a platform with over a thousand available NLP datasets, and the closest we found was the BBC News Articles dataset. After applying the compatibility by equation (1), only approximately 57% of the words needed were presented, which we considered not high enough.

The alternative was to use web scraping algorithms to acquire more unlabeled data from the same sources, which would give a higher chance of providing compatible texts. The original dataset had 357 entries, with 260 of them labeled. We obtained 518 new data elements with the second scraping. When comparing the 260 original elements to the 518 new unlabeled ones, we achieved over 75% compatibility, indicating the alternative assumption was correct. Table 2 displays the result of the comparisons made. Compatibility means the percentage of the original dataset that is contained in the new dataset.

Table 2: Compatibility results (\*base = labeled MCTI dataset entries).

Dataset	Compatibility to base*
Labeled MCTI	100%
Full MCTI	100%
BBC News Articles	56.77%
New unlabeled MCTI	75.26%

Figure 6 shows a representation of the weights trained by Word2Vec. It is obtained through dimensionality reduction using t-Distributed Stochastic Neighbor Embedding (t-SNE). In *a*), the training was done using the 260 labeled data; in *b*), it was used 875 unlabeled data. In the image, it is possible to observe that the concentrated/dense area increased considerably with the addition of the new dataset and its tokens, demonstrating the enrichment of the weights concerning the desired contextual information. If pre-training was done with an incompatible dataset, it would be likely that other large dense areas would appear and probably interfere with the meaning of the words.

## 5.2 Model Training with Word2Vec Embeddings

Now we have a pre-trained model of word2vec embeddings that has already learned relevant meanings for our classification problem. We can couple it to our classification models (Fig. 4), realizing transfer-learning and then training the model with the labeled data in a supervised manner. The new coupled model can be seen in Figure 5 under word2vec model training.



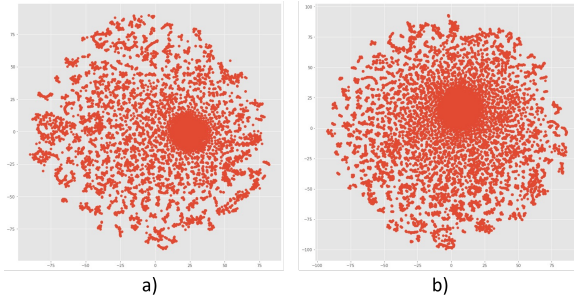


Figure 6: Pre-trained weights by Word2Vec: a) weights trained with labeled MCTI data; b) weights trained with Full MCTI + New unlabeled MCTI data.

The Table 3 shows the obtained results with related metrics. With this implementation, we achieved new levels of accuracy with 86% for the CNN architecture and 88% for the LSTM architecture.

Table 3: Results from Pre-trained WE + ML models.

ML Model	Accuracy	F1 Score	Precision	Recall
NN	0.8269	0.8545	0.8392	0.8712
DNN	0.7115	0.7794	0.7255	0.8485
CNN	0.8654	0.9083	0.8486	0.9773
LSTM	0.8846	0.9139	0.9056	0.9318

### 5.3 Transformer-based Implementation

Another way we used pre-trained vector representations was by use of a Longformer (Beltagy et al., 2020). We chose it because of the limitation of the first generation of transformers and BERT-based architectures involving the size of the sentences: the maximum of 512 tokens. The reason behind that limitation is that the self-attention mechanism scale quadratically with the input sequence length  $O(n^2)$  (Beltagy et al., 2020). The Longformer allowed the processing sequences of a thousand characters without facing the memory bottleneck of BERT-like architectures and achieved SOTA in several benchmarks.

For our text length distribution in Figure 3, if we used a Bert-based architecture with a maximum length of 512, 99 sentences would have to be truncated and probably miss some critical information. By comparison, with the Longformer, with a maximum length of 4096, only eight sentences will have their information shortened.

To apply the Longformer, we used the pre-trained base (available on the link) that was previously trained with a combination of vast datasets as input to the model, as shown in figure 5 under Longformer model training. After coupling to our classification models, we realized supervised training of the whole model. At this point, only transfer learning was applied since

more computational power was needed to realize the fine-tuning of the weights. The results with related metrics can be viewed in table 4. This approach achieved adequate accuracy scores, above 82% in all implementation architectures.

Table 4: Results from Pre-trained Longformer + ML models.

ML Model	Accuracy	F1 Score	Precision	Recall
NN	0.8269	0.8754	0.7950	0.9773
DNN	0.8462	0.8776	0.8474	0.9123
CNN	0.8462	0.8776	0.8474	0.9123
LSTM	0.8269	0.8801	0.8571	0.9091

## 6 DISCUSSION

In this section, we will discuss the problem, the models used and the results obtained in the context of classification in NLP. Table 5 shows the overview of the final results obtained by the experiments:

Table 5: Comparison of the results of three scenarios.

Model	Accuracy	F1 Score	Precision	Recall
Baseline				
Bag of words	0.86	0.85	0.85	0.85
Key words	0.7808	0.7802	0.9358	0.6711
Keras Word-embedding				
NN	0.8269	0.8620	0.8212	0.9129
DNN	0.8269	0.8650	0.8952	0.8447
CNN	0.8462	0.8756	1.0000	0.7803
LSTM	0.8269	0.8675	0.8276	0.9129
Pre-trained Word2Vec Embeddings				
NN	0.8269	0.8545	0.8392	0.8712
DNN	0.7115	0.7794	0.7255	0.8485
CNN	0.8654	0.9083	0.8486	0.9773
LSTM	0.8846	0.9139	0.9056	0.9318
Pre-trained Longformer				
NN	0.8269	0.8754	0.7950	0.9773
DNN	0.8462	0.8776	0.8474	0.9123
CNN	0.8462	0.8776	0.8474	0.9123
LSTM	0.8269	0.8801	0.8571	0.9091

When analyzing the data, we can see an aspect that hinders the visualization of the accuracy improvements: the values changes in 1.9% jumps. The amount of labeled testing data is minimal, and since we carry out 80-20 training segmentation, we only verify 52 items, which explains the jumps.

It is possible to verify that most the tested classification models showed improved accuracy after pre-training. In particular, the network composed of pre-trained word2vec embeddings + LSTM network obtained the best result for the target dataset of 88% accuracy and over 90% precision.

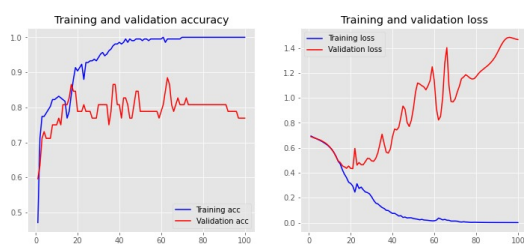


Figure 7: LSTM model with Word2Vec training curve.

Taking a deeper look at the W2V + LSTM training curve in Figure 7, we can see that the model reached its best accuracy around the 60th epoch. We can also see constant fluctuations in the accuracy values, which are very common in LSTM models. After that, even though the training accuracy kept improving, there was a pattern of overfitting where the validation accuracy declined.

The Longformer results did surpass the simple embeddings model and, although satisfactory, can probably be improved by fine-tuning the model. That will require more computing power and optimizations and will be done in further studies.

## 7 CONCLUSION AND FUTURE WORK

Advances in artificial intelligence in recent years have aroused the interest of MCTI in automating the process of identifying and selecting relevant opportunities for Brazilian projects. This task plays a significant role in helping to promote scientific research and in its funding.

This research has developed machine learning-based methods to automatically process the collected text for the Financing Products Portfolio (FPP) organized by the Brazilian Ministry of Science, Technology, and Innovation (MCTI). The main objective consists of performing automated searches on known digital platforms using data scraping techniques to collect information about opportunities. Then, these data will serve as input for an algorithm that uses NLP methods to classify opportunities in terms of eligibility for Brazilian projects.

In this article, we reported the following achievements on the first two steps of the project, mentioned in section 2.1:

- presentation of the collected data, including the texts of financing opportunities provided by many R&D funding organizations worldwide.
- presentation of the performance by the previous study, which established the accuracy rate of 86%

in the text classification as the baseline.

- the achievement of the accuracy rate of 84% by the solution using Word Embedding and Deep learning such as NN, DNN, CNN, and LSTM.
- the achievement of the accuracy rate of 88% by the solution of pre-training Word2Vec embeddings with transfer learning plus deep learning models such as NN, DNN, CNN, and LSTM.
- the achievement of the accuracy rate of 84% with the pre-trained Longformer plus deep learning models such as NN, DNN, CNN, and LSTM.

All the data and models used here, as well as the results obtained, are publicly available in github<sup>1</sup>.

It is worth mentioning the comparative contribution of this research. It is well-known that Google, Microsoft, IBM, OpenAi, and other prominent international AI companies have human, data, and computing resources that allowed them to develop well-known machine learning methods and theories such as Transformer, GPT, BERT, and Vision Transformer. In our case, however, we are faced with limited resources and small-scale structured data. We introduced the pre-learning strategy to learn the features from larger-scale unstructured data and then used transfer learning methods to improve the accuracy of data classification through deep learning. Achieving the same goal with limited resources is an important initiative and is an essential measure of technical progress in developing countries, especially in the development of AI technology.

Apart from the benefit of automating the process of identifying and selecting opportunities that already exist today (which is the focus of this research), we can also emphasize that the technology developed here may be applied in several fields of public management for other related sectors from federal or state government. In particular, the proposed solution can benefit other research projects in developing countries, which face similar challenges with limited resources and small-scale structured data.

## ACKNOWLEDGEMENTS

The Brazilian Ministry of Science, Technology, and Innovation (MCTI) has partially supported this project. We sincerely thank Dr. Joao Gabriel Souza, who leads the dataset construction and kindly shared the data for this study.

<sup>1</sup><https://github.com/chap0lin/WEBIST2022>

## REFERENCES

- Ainslie, J., Ontanon, S., Alberti, C., Cvícek, V., Fisher, Z., Pham, P., Ravula, A., Sanghai, S., Wang, Q., and Yang, L. (2020). Etc: Encoding long and structured inputs in transformers.
- Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., and Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques.
- Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Chowdhury, S. A., Stepanov, E. A., and Riccardi, G. (2016). Predicting User Satisfaction from Turn-Taking in Spoken Conversations. In *Proc. Interspeech 2016*, pages 2910–2914.
- Church, K. W., Chen, Z., and Ma, Y. (2021). Emerging trends: A gentle introduction to fine-tuning. *Natural Language Engineering*, 27(6):763–778.
- Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.
- Do, C. B. and Ng, A. Y. (2005). Transfer learning for text classification. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press.
- Fei-Fei, L., Fergus, R., and Perona, P. (2003). A bayesian approach to unsupervised one-shot learning of object categories. In *proceedings ninth IEEE international conference on computer vision*, pages 1134–1141. IEEE.
- Gers, F. A., Schmidhuber, J., and Cummins, F. (2000). Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471.
- Gron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, Inc., 1st edition.
- Han, J. and Kamber, M. (2000). Data mining: Concepts and techniques.
- Hazen, T. J. (2011). Mce training techniques for topic identification of spoken audio documents. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(8):2451–2460.
- Heaton, J. (2020). Applications of deep neural networks. *arXiv preprint arXiv:2009.05673*.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.
- IBM Cloud Education (2020). What are convolutional neural networks? <https://www.ibm.com/cloud/learn/convolutional-neural-networks>. IBM Cloud Learn Hub.
- Kesiraju, S., Burget, L., Szőke, I., and Černocký, J. (2016). Learning document representations using subspace multinomial model. In *Proceedings of Interspeech 2016*, pages 700–704. International Speech Communication Association.
- Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.
- Kontonatsios, G., Spencer, S., Matthew, P., and Korkontzolos, I. (2020). Using a neural network-based feature extraction method to facilitate citation screening for systematic reviews. *Expert Systems with Applications: X*, 6:100030.
- Kowsari, Meimandi, J., Heidarysafa, Mendu, Barnes, and Brown (2019). Text classification algorithms: A survey. *Information*, 10(4):150.
- Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., Gerber, M. S., and Barnes, L. E. (2017). HDL-Text: Hierarchical deep learning for text classification. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE.
- Lilleberg, J., Zhu, Y., and Zhang, Y. (2015). Support vector machines and word2vec for text classification with semantic features. In *2015 IEEE 14th International Conference on Cognitive Informatics Cognitive Computing (ICCI\*CC)*, pages 136–140.
- Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B., and Bengio, Y. (2017). A structured self-attentive sentence embedding.
- Luque, J., Segura, C., Sánchez, A., Umbert, M., and Galindo, L. A. (2017). The role of linguistic and prosodic cues on the prediction of self-reported satisfaction in contact centre phone calls. In *Proc. Interspeech 2017*, pages 2346–2350.
- McCann, B., Bradbury, J., Xiong, C., and Socher, R. (2017). Learned in translation: Contextualized word vectors. In *NIPS*.
- Meinzer, S., Jensen, U., Thamm, A., Hornegger, J., and Eskofier, B. (2017). Can machine learning techniques predict customer dissatisfaction? a feasibility study for the automotive industry. *Artif. Intell. Res.*, 6:80–90.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS’13*, page 3111–3119, Red Hook, USA. Curran Associates Inc.
- Miller, E. G., Matsakis, N. E., and Viola, P. A. (2000). Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 464–471.
- Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., and Gao, J. (2021). Deep learning-based text classification: A comprehensive review. *ACM Comput. Surv.*, 54(3).
- Mohammed, A. H. and Ali, A. H. (2021). Survey of BERT (Bidirectional Encoder Representation Transformer)

- types. In *Journal of Physics: Conference Series*, volume 1963, page 012173. IOP Publishing.
- Nielsen, M. A. (2015). *Neural networks and deep learning*, volume 25. Determination press San Francisco, USA.
- Pan, S. J., Tsang, I. W.-H., Kwok, J. T.-Y., and Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22:199–210.
- Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.
- Pappagari, R., Villalba, J., and Dehak, N. (2018). Joint verification-identification in end-to-end multi-scale cnn framework for topic identification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203.
- Park, Y. and Gates, S. C. (2009). Towards real-time measurement of customer satisfaction using automatically generated call transcripts. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09*, page 1387–1396, New York, USA. Association for Computing Machinery.
- Parloff, R. (2016). Why deep learning is suddenly changing your life. *Fortune. New York: Time Inc.*
- Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Int. Conf. on Machine Learning - Volume 28, ICML'13*, page III–1310–III–1318. JMLR.org.
- Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations.
- Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.
- Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A. Y. (2007). Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the 24th Int. Conf. on Machine Learning, ICML '07*, page 759–766, New York, USA. Association for Computing Machinery.
- Ruder, S., Peters, M. E., Swayamdipta, S., and Wolf, T. (2019). Transfer learning in natural language processing. In *Proceedings of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, USA. Association for Computational Linguistics.
- Seo, M., Min, S., Farhadi, A., and Hajishirzi, H. (2017). Neural speed reading via skim-rnn.
- Shen, D., Zhang, Y., Henaio, R., Su, Q., and Carin, L. (2018). Deconvolutional latent-variable model for text sequence matching. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).
- Silva, B., Alves, J., Rebeschini, J., Querol, D., Pereira, E., and Celestino, V. (2021). Data science applied to financial products portfolio. In *Analys of Meeting of National Association of Post-graduation and Research in Administration*.
- Sree, K. and Bindu, C. (2018). Data analytics: Why data normalization. *International Journal of Engineering and Technology (UAE)*, 7:209–213.
- Sun, C., Qiu, X., Xu, Y., and Huang, X. (2019). How to fine-tune bert for text classification? In Sun, M., Huang, X., Ji, H., Liu, Z., and Liu, Y., editors, *Chinese Computational Linguistics*, pages 194–206, Cham. Springer International Publishing.
- Thompson, N. C., Greenewald, K., Lee, K., and Manso, G. F. (2020). The computational limits of deep learning.
- van den Bulk, L. M., Bouzembrak, Y., Gavai, A., Liu, N., van den Heuvel, L. J., and Marvin, H. J. (2022). Automatic classification of literature in systematic reviews on food safety using machine learning. *Current Research in Food Science*, 5:84–95.
- van Dinter, R., Catal, C., and Tekinerdogan, B. (2021). A decision support system for automating document retrieval and citation screening. *Expert Systems with Applications*, 182:115261.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.
- Wang, G., Li, C., Wang, W., Zhang, Y., Shen, D., Zhang, X., Henaio, R., and Carin, L. (2018). Joint embedding of words and labels for text classification.
- Weigang, L. (1998). A study of parallel self-organizing map. *arXiv preprint quant-ph/9808025*.
- Weigang, L. and da Silva, N. C. (1999). A study of parallel neural networks. In *IJCNN '99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, volume 2, pages 1113–1116. IEEE.
- Weigang, L., Enamoto, L. M., Li, D. L., and Rocha Filho, G. P. (2022). New directions for artificial intelligence: Human, machine, biological, and quantum intelligence. *Frontiers of Information Technology & Electronic Engineering*, 23(6):984–990.
- Xiao, L., Wang, G., and Zuo, Y. (2018). Research on patent text classification based on word2vec and lstm. In *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, volume 01, pages 71–74.
- Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.
- Yogatama, D., Dyer, C., Ling, W., and Blunsom, P. (2017). Generative and discriminative text classification with recurrent neural networks.
- Zhang, D., Xu, H., Su, Z., and Xu, Y. (2015). Chinese comments sentiment classification based on word2vec and svmperf. *Expert Systems with Applications*, 42(4):1857–1863.

Zhang, L., Lu, L., Nogues, I., Summers, R. M., Liu, S., and Yao, J. (2017). Deeppap: Deep convolutional networks for cervical cell classification. *IEEE Journal of Biomedical and Health Informatics*, 21(6):1633–1643.

