Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 207 (2022) 1861–1870

Procedia Computer Science

www.elsevier.com/locate/procedia

26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022)

Automatic Evaluation of French Research Projects in the Acquisition Process of Research Tax Credit (CIR)

J. Carvallo[a], Z. Ramezanpanah[a], A. Rodriguez[a]

[a]*MYTEAM, 179 rue de la Pompe, 75116, Paris, France*

Abstract

In this work, we evaluated research projects of French companies using Natural Language Processing. To this end, we designed a system able to estimate the probability of obtaining a research tax credit (CIR) for a project based on its technical description. This system is designed around two modules whose outputs are concatenated and fed to a fully-connected neural network that predicts the probability of success for the project. The first module uses the FastText algorithm and a Convolutional Neural Network to extract a Text Embedding vector. The second module uses an unsupervised knowledge graph extraction method and a Graph Neural Network to extract a Graph Embedding vector. The texts used as data in this study describe the research projects of companies and are written in French. Due to their high confidentiality, no similar examples exist in the literature. This data is provided by a partner consulting firm whose work consists in helping companies raise funds for their research projects. Since the methods in the literature were not effective in extracting the knowledge graphs used in the second module for our data, we present a new Knowledge Graph extraction using an unsupervised Named Entities Recognition, NER, as our contribution.

Keywords: Text classification; Unsupervised NLP; Knowledge Graph; French technical descriptions; Text embedding.

1. Introduction

Daily advances in machine learning and artificial intelligence give researchers ideas of replacing humans with machines in many fields. Text classification is one of these domains and consists of building learning systems taking texts as input and producing a text representation and a label as output. Today, this technology plays a vital role in many jobs and businesses, such as organizing documents, categorizing news topics, sentiment analysis, advertising industry, diagnosing diseases using text, etc [29, 41, 28].

E-mail address: joachimcarvallo@myteam.ai

1877-0509
10.1016/j.procs.2022.09.244

The first step in text classification is to select important features to describe the text. This was usually done with a handmade feature vector for a given piece of text in the past [37]. Today, due to the remarkable success of the word embedding learning method, a piece of text can be represented by an embedding learned automatically from raw text using neural networks. Learning methods often include sequence-based learning models [5, 6, 43] and graph-based learning models [40, 4, 41]. These learning methods have been highly regarded for classifying texts in various fields and have been used extensively in texts written in English. The abundance of data in various fields such as finance [21], business [12], medicine [16], and daily events [38] has made it easy to use NLP to analyze and even classify texts. Despite all these researches and data in various fields, the lack of textual data in languages other than English and highly technical fields limits the use of machine learning. One of these limitations is the construction of supervised training data, which in addition to data, requires the use of specialized knowledge in these fields. Therefore, the use of text mining technology in technical texts for the direct use of machine/deep learning is associated with many complexities.

In this paper, we evaluated the texts describing the activities of French companies, often start-ups. These texts are highly specialized, and the methods available in the literature cannot recognize the entities, the relationships between them, and understand the texts. Therefore, our contribution to this study is to present a new method for identifying entities that improve the extracted knowledge graph.

The rest of this paper is structured as follows:
Section 2 presents an overview of the current state of the art. Section 3 describes the data sets used in this study in detail. In Section 4, we first provide an overview of the pipeline and then describe the methods used in it. Finally, in Sections 5 and 6, respectively, we review the numerical results obtained and the work to be done to improve this research in the future.

2. State of the art

With the recent success of Word Embeddings techniques [24, 29], sentences are often represented as sequences of features in NLP tasks. Hence, standard deep learning techniques such as recurrent neural networks [32] have been widely used to model text sequences [42, 2]. The methods used in the literature often require labeled data-sets to be trained. Many data-sets in various fields, especially in English, are available to everyone in open source. Among these, we can mention the data-sets of GLUECoS [15] for Code-Switched NLP, British National Corpus (BNC) [19], Movie reviews with one sentence per review [26], Corpus du Français Parlé Parisien des années 2000 (CFPP2000) [3] for French and Stock Values and Earnings Call Transcripts [31] for Sentiment Analysis in English texts. The existence of these data-sets led to much research being done with the well-known methods of machine learning. For example, in [5], the authors used a Convolutional Neural Network (CNN) to classify texts in MR[26] data-set. In [41], the authors used a graph neural network (GNN) to classify texts of various data-sets such as *R8*, *R52*, and *20newsgroups*. Despite dozens of other benchmarks in different languages, the lack of data-sets in specialized fields, mainly non-English texts, is still very noticeable. Lack of data and consequent inability to use supervised methods led researchers to use semi-supervised and unsupervised methods. These methods are often used for various tasks in the text classification process, such as Named Entity Recognition (NER), Part of Speech (POS), and parsing. The lack of proper performance of NER and POS causes the failure of text classification algorithms because they cannot understand texts without proper knowledge of the entities and the relationships between them. Some web encyclopedias, such as Wikipedia, use semi-supervised methods to generate NER training data. The main idea of this method is to convert links to labels by categorizing Wikipedia pages into different types of entities. Some methods classify web pages based on handcrafted rules that use Wikipedia classification information [30]. These rule-based methods have high accuracy but low coverage. To achieve better performance, the authors in [25] have used a classifier that additional hand-tagged Wikipedia pages have trained with a variety of entities. Bidirectional Encoder Representations from Transformers (BERT) is another significant achievement in text classification, designed to pre-train deep bidirectional representations of unlabeled text [9]. In researches, based on unsupervised methods, BERT attracted a great deal of attention [17, 36, 27]. In order to reproduce and validate the results obtained so far only for English, the authors in [23] use the new OSCAR multilingual collections [34] to train a monolingual language model for French, called CamemBERT. CamemBERT was widely used in the literature because of its ability to understand French texts

[35, 22, 1]. Because of CamemBERT's effectiveness in French texts, we also used it for an unsupervised NER task, constituting a brick of the knowledge graph generation method that we present in this work.

3. Data set

The research tax credit (CIR) is a French subsidy to support the R&D activities of companies without the restriction of sector or size. The companies that engage in fundamental research and experimental development can benefit from the CIR by deducting a portion of these expenses from their taxes. The innovation tax credit (CII) is an extension of the CIR for projects with a lower level of research and technical expertise. These projects, qualified as innovation projects, focus on subjects closer to existing technologies. An essential part of our partner's activity is to support companies in their application process for CIR and CII. The consultants guide companies in applying for these credits and assist them in drafting documents describing their activities in the proper uniform format. After more than eight years of existence, the company has an essential history of these technical files describing research and development projects. These documents are typically organized in several parts, depending on whether the objective is to obtain CIR or CII: Presentation of the company and the R&D team, the project's objectives, economic context, scientific uncertainties & technological barriers, state of the art, description of the work carried out. In this work, we focused on the only part systematically included and uniformed for CIR and CII applications: the description of the work carried out. From this description, our objective is to classify the projects between CIR and CII, which is an excellent proxy for the level of technicality of the research conducted. Our data set consists of 489 technical descriptions. Among them, 173 received CII, and the remaining 316 received CIR. All graphics included in these texts have been removed to focus on the NLP aspect of the problem. A text consists of an average of 2 561 tokens, which represents approximately five pages, and 80% of the texts are from 846 to 4 752 tokens long (i.e., from one to ten pages). However, there are extreme values: the shortest text contains only 201 tokens, and the most extended text contains 15 509 tokens.

4. Methodology

The system presented in this section has been finalized after many trials and errors with various strategies and algorithms. All the choices made for its design and architecture have been in line with the ultimate goal of this work. Figure 1depicts the architecture of our system. As shown in this figure, this system consists of two modules in parallel, which are described below. The aim of creating two distinct modules is to extract different and complementary information from the texts. On the one hand, the first module aims at focusing on elements rather related to the form and the writing of the text: vocabulary employed, recurrence of certain words, sentence structures, etc. On the other hand, the second module aims at focusing on elements rather related to the content and semantics of the text: how the different concepts introduced are organized and influence each other. Aspects of content and form are both fundamental to correctly classifying our documents between our two categories.

4.1. Text Embedding Module

4.1.1. FastText Word Embedding

The first step of this module is to transform the words of our texts into vectors using a lexical embedding algorithm [8] using Fast Text library [10]. Fast Text is a derivative of the Word2Vec algorithm [13] and provides a word representation of texts from a large amount of unstructured text. In this paper, we used the body of French texts CoFiF [7] which contains more than 188 million tokens in 2655 reports that cover reference documents, annual and quarterly test reports. The data was collected by 60 large French companies listed on the main French stock indices CAC40 and CAC Next 20. This collection covers more than 20 years, from 1995 to 2018. As shown in Figure 1, the input of this module is a text file, which is tokenized into words. Each word is mapped to a word vector representation, i.e., a word embedding, such that an entire text file will be mapped to a matrix of size $n \times m$, where n is the number of words in the whole text and m the dimension of the embedding space (in our case $m = 100$).

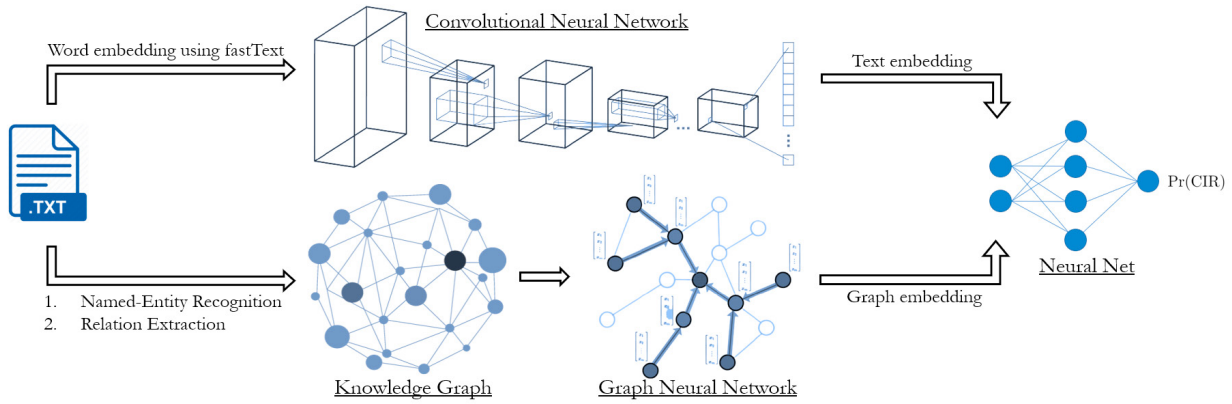

Fig. 1. Designed system architecture.

4.1.2. Convolutional Neural Network

Convolutional Neural Network (CNN) is a class of artificial neural networks, most often used for image analysis. Like almost all other neural networks, they are trained with the backpropagation algorithm but differ in their architecture. CNN are designed to recognize visual patterns directly from pixel images with minimal preprocessing. Since the introduction of their basic concepts in [11] and their democratization with [18], they have also been successfully used in recommender systems or natural language processing.

In this step, we trained a one-dimensional CNN. The input of this network is the $n \times m$ matrix obtained in Section 4.1.1. We then apply several convolution operations of various sizes to this matrix. Each convolution is a filtering matrix, $fm \in R^{s \times m}$ where s is the size of the convolution. The convolution operation is obtained by:

$$c_i = f(\sum_{j,k} fm_{j,k}(X_{[i:i+h-1]})_{j,k} + b) \tag{1}$$

In which $b \in R$ is called the bias term and $f(x)$ in a non-linear function. In this paper, we chose relu as it results in faster training without usually making a significant difference in accuracy. The output of this equation, c_i is a concatenation of the convolution operator on all possible word windows in the input text file. We also used a wide convolution [14] due to the use of the zero-padding strategy. We then performed a max-pooling operation to each convolution $c_{max} = max(c)$. This operation extracts the most important feature in each convolution. In this work, we have 100 of size 3 and 100 of size 7. The values obtained for each of the filters in the model are aggregated in order to give a final representation of our text, of the same dimension for all the texts, which will serve as input for a layer of neurons carrying out the final classification.

4.2. Graph Embedding Module

The idea of this module is to focus more specifically on the semantics of our texts. The method we propose is to extract a knowledge graph from each text and then analyze it with a GNN to obtain an embedding of the information contained in each text. The generation of the knowledge graph is divided into two parts. (1) Named Entity Recognition identifies the essential concepts in our text. (2) Relation Extraction identifies the logical links between our concepts.

4.2.1. Named Entity Recognition (NER)

The objective of the NER is to identify and classify the entities in our text, which will then form the nodes of our knowledge graph. Usually, identifying entities requires labeled data to train NER machine learning models. The data used in this study differ from the existing NER data set in two significant respects. First, our texts are highly technical, so the entities we seek to identify are different from ordinary entities such as individuals, places, or organizations. Second, the number of entity classes in most NER data sets is less than six. We need more entity classes to obtain

relatively complex graphs (the number of entity classes corresponds to the maximum number of nodes for our knowledge graph). Therefore, we cannot directly use pre-trained models nor train our models on available NER data sets. To tackle this NER problem, we divided it into two parts: identifying entities in our texts and classifying those entities.

We annotated a sample of our texts for the identification part and fine-tuned the "fr_core_news_lg" model from the Spacy library with these data. Directly adding the classification part to this model was impossible because annotating our texts correctly with many possible entities is a challenging task, even for humans. We then opted for an unsupervised classification strategy based on the Camembert model. The first step in this method is to create a list of meaningful entity classes for our task. We used the pretrained *"camembert-base"* model [23] to obtain context-dependent representations of our words and the pretrained *"french-camembert-postag-model"* to obtain their Part of Speech (POS) tags and to keep only common and proper nouns as possible entities in our texts. We also removed tokens from the list that did not exist in the OpenLexicon [20] data set, which is an almost complete data set of French words. This ensured that only correct nouns were kept and avoided keeping sub-parts of nouns that the Camembert tokenizer would have cut off. For the remaining tokens, we used the last layer of the Camembert model as a feature vector so that each token was associated with a vector with 768 features. Using Algorithm 1, we came up with a list of 50 sub-lists, each of which represents an entity class. This algorithm aims to cluster the set of nouns used in our texts with clusters that can have non-zero intersections. Each of these clusters will then constitute an entity class. This number of 50 classes is a hyper-parameter of our model.

The next step in this section is to classify the entities from new texts. We first labeled the nouns in the list of entities according to their cluster, i.e., the number of their sub-lists. In this case, if $entities = [e_0, e_1, \ldots, e_i, \ldots, e_{49}]$ is our list in which each e_i, $0 \leq i \leq 49$ is a sub-list, then all the tokens in the e_i will be labeled with i. We have a data set consisting of 1642 tokens labeled between 0 and 49. As mentioned at the beginning of this section, using existing NER methods has resulted in almost 90% of the entities in the MISC, Miscellaneous, category. To evaluate the introduced NER algorithm, we labeled the entities in our data by calculating the cosine similarity of their last Camembert layer with these 50 entities. We then randomly split these entities into a train and a test set with a ratio of 70%-30%. Afterward, we trained a logistic regression [39] model. This model correctly classified 75% of the test entities[1].

4.2.2. Relation Extraction (RE)

In this section, we seek to extract the relationships between the entities that we identified and classified in the previous step. We used a rule-based approach built on tags of part of speech and syntactic dependencies extracted by Spacy's pre-trained module. The function of this system is to extract one or more triplets in the form of (subject, verb, complement) in all sentences of the input text. In the first step, we identify the verb (or verbal group) and the possible associated negation. Afterward, we look for the subject associated with the identified verbs and examine whether it is part of an entity class. Finally, we look for the object related to the identified subject and verb and then examine whether it is part of an entity. Since the pattern of the verb, subject, and object is not always observed in the written text (for example, in bulleted sentences), there were errors in identifying these triplets. To solve this problem, we defined rules that deal with several specific cases, such as sentences with several verb-object groups and a single subject or enumerations of the form: "_: _ ; _ ; _.". At this point, we have extracted relationships, but the number of possible labels for the edges is very high. So just as we reduced the number of entities to reduce the complexity and cost of calculations, we also reduced the number of possible relationships. For this purpose, we manually defined nine classes, each containing a certain number of verbs represented in our data. Then each new verb (i.e., edge label) is assigned to the nearest class with a cosine similarity in the FastText embedding space. Finally, we double the number of relation classes to account for negations: we add a symmetrical negative class for each of the nine classes. If there is a negation in the sentence from which the relation comes, then the assigned relation is the negative class of the nearest class in the FastText embedding space. These 9 possible relations are : "être", "développer/permettre", "avoir", "proposer/offrir", "réaliser/permettre", "présenter/illustrer", "analyser/étudier", "nécessiter/devoir", "implémenter/calculer" and their 9 negative versions.

[1] The data used is private, and we are not allowed to publish it, but the Python code of this algorithm, without input, can be found here

Algorithm 1: Unsupervised NER algorithm.

Data: ***Texts*** : a list of all text file in training data
Result: ***Entities*** : a list of all recognized entities

```
final_txt ← concatenate(Texts);
list_of_tokens ← camembert_tokenizer(final_txt);
POS ← french-camembert-postag(list_of_tokens);
camembert_list ← camembert-base(list_of_tokens);
for token in list_of_tokens do
    if POS(token)=="NC" or POS(token)=="NPP"        ▷ NC: nom commun, NPP: nom propre then
        if token is in OpenLexicon then
            noun_list + = (token, camembert_list(token))
grouped_noun_list ← group_by_noun(noun_list);
camembert_list ← mean(grouped_noun_list);
for (vect_1, vect_2) in all possible individual pairs of camembert_list do
    cos_sim_list + = cosine_similarity(vect_1, vect_2)
grouped_cos_sim_list ← Grouping words that have a cosine similarity greater than 70%
grouped_cos_sim_list_sorted ← Sort by sub-list size and in descending order
drop_list ← [ ];
n ← size(grouped_cos_sim_list_sorted);
for i in 1:n do
    if list_i is in drop_list then
        pass
    else
        for j in (i+1):n do
            if Number of common words of (list_i, list_j) > 45% of the length of list_j then
                drop_list + = list_j
Delete repeated list in drop_list
Entities: grouped_cos_sim_list_sorted - drop_list
```

Below is an example of the entire process:

1. Identification of entities:
   Ce mécanisme permet de protéger les données internes critiques.
2. Relation Extraction:
   (Ce mécanisme", permet de protéger, les données internes critiques)
3. Entities and relation classification:
   (processus, réaliser/permettre, objectif)

After extracting relations from each sentence of our text, we obtain a knowledge graph representing our text. Figure 2 shows an example of a knowledge graph that we extract with this method from a short text of our data.

4.2.3. Graph Neural Network

In the final stage of this module, we used a Graph Neural Network (GNN) to extract an embedding from our knowledge graph. More specifically, we used two edge-conditioned convolutional layers [33] of size 32 and a global average pooling layer.

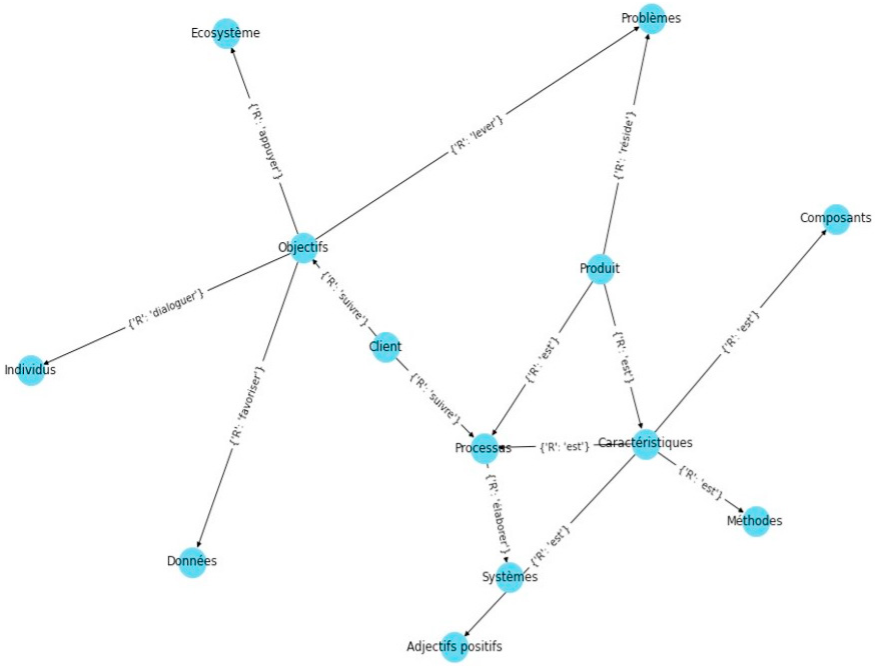

Fig. 2. An example of a knowledge graph extracted on a short text.

4.3. Prediction

After obtaining a Text Embedding vector using the first module and a graph embedding vector using the second module, we concatenated them to represent our text fully. Thereupon, we used a fully-connected neural network taking as input the concatenation of the embedding vectors and giving our variable to predict as output.

5. Results

Table 1. Performance comparison between the different modules.

Module	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Graph Embedding Module	79.7 ± 7.8	83.4 ± 9.4	84.1 ± 8.4	83.4 ± 7.2
Text Embedding Module	93.4 ± 6.5	95.4 ± 7.0	94.7 ± 8.3	94.6 ± 5.2
Full model	96.1 ± 3.7	97.4 ± 3.2	96.8 ± 4.8	97.0 ± 2.7

In this section, we discuss the results obtained from this system. We present the selected hyperparameters for training the model in Table 2. As described in Section 3, our data is not a balanced data set, so in addition to accuracy, we also calculated F1-Score, Precision, and Recall criteria. We used 10-fold cross-validation to compute the performances reported in 1, in the format $\mu \pm \sigma$ where μ is the mean value of the metric over the ten folds and σ the standard deviation. We can see that both modules perform well individually. Therefore they each can extract useful information from the text. The Text Embedding Module is the better performing of the two. The full model performs better than the individual modules, which indicates that the two modules provide valuable and complementary information for the prediction. Figure 3 shows the average confusion matrix for the full model with 10-fold cross-validation. We find that the errors are balanced between our two classes, so no class is more problematic than the other. Finally, we obtained an overall accuracy of 96.1% on the task, which is very satisfactory as it is close to 100%.

Table 2. Best hyperparameters.

nb_filters: 100	filter_size_a : 3	filter_size_b: 7	batch_size: 8	epochs: 50	my_optimizer:'adam	learning_rate: 10^{-3}
l2_reg: 0.05	dropout_rate: 0.4	decay_rate: 0.99	loss: Binary Cross Entropy			

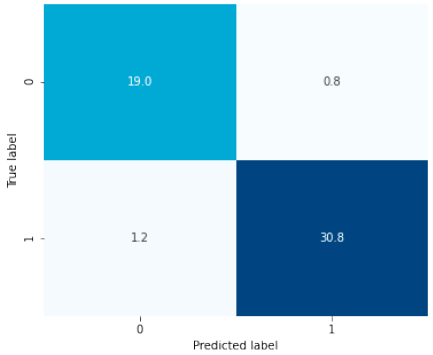

Fig. 3. Average confusion matrix for the full model with 10-fold cross validation.

6. Discussion, Conclusion, and Future Works

In this section, we discuss the results obtained and our contributions. Then we present the steps needed to improve and expand the results of this article.

6.1. Discussion and Contribution

This study aims at analyzing technical texts using natural language processing. These texts describe the projects of French companies. The proposed system in this paper can determine the type of project (research or innovation). In order to achieve this goal, the system uses two modules, which are called Text Embedding and Graph Embedding, respectively. In the first module, a text file is converted to a matrix using the Fast Text algorithm, serving as input to a single-layer CNN. The output of this module is a vector called Text Embedding Vector. In the second module, the first problem we encountered was our texts' high specialization and uniqueness. To our knowledge, these data have no similar examples in the literature to date. The first consequence of this problem was the classification of entities, so that approximately 90% of the entities were in the category of MISC, meaning Miscellaneous entities. Incorrect classification of these entities led to the failure to construct meaningful knowledge graphs. So, our first challenge was to create an algorithm to perform Named Entity Recognition. This algorithm, which is a new method of unsupervised NER, is our main contribution to this paper and consists of several steps that we described in Algorithm 1. The output of this algorithm was a list of 50 entities. To test the performance of our contribution, we trained a logistic regression model with 70% of the data and evaluated its efficiency with the remaining 30%. The result of the trained model indicated that this time 75% of the entities were correctly classified. This unsupervised NER method, combined with our rule-based relationship extraction algorithm, completed the requirements to extract meaningful knowledge graphs. Finally, we used a GNN to extract an embedding of our knowledge graphs, constituting the output of the second module, called Graph Embedding Vector. By concatenating the two output vectors of these two modules and training a fully-connected neural network, we were able to classify the evaluation data with 96.1% accuracy.

6.2. Future works

This paper presents a promising first stage of our research. We divided the future works following this study into several steps. The first step to improving the system is to solve the over-fitting problem. We can solve this problem by collecting more data, and we are now looking for other partners to scale up our data set. At the same time, we

are trying to use this data to approximate the budget necessary to conduct a research project. This task is challenging and, for now, associated with a high error as we do not currently have enough data. However, with more data, we will publish our results for budget estimation.

7. Acknowledgment

This research was supported by MYTEAM. We thank our colleagues who provided data and expertise that greatly assisted the research.

References

[1] Bailly, A., Blanc, C., Guillotin, T., 2021. Classification multi-label de cas cliniques avec camembert (multi-label classification of clinical cases with camembert), in: Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Atelier DÉfi Fouille de Textes (DEFT), pp. 14–20.
[2] Berglund, M., Raiko, T., Honkala, M., Kärkkäinen, L., Vetek, A., Karhunen, J.T., 2015. Bidirectional recurrent neural networks as generative models. Advances in neural information processing systems 28.
[3] Branca-Rosoff, S., Fleury, S., Lefeuvre, F., Pires, M., 2000. Discours sur la ville. Corpus de français parlé parisien des années 2009.
[4] Cai, H., Zheng, V.W., Chang, K.C.C., 2018. A comprehensive survey of graph embedding: Problems, techniques, and applications. IEEE Transactions on Knowledge and Data Engineering 30, 1616–1637.
[5] Chen, Y., 2015. Convolutional neural network for sentence classification. Master's thesis. University of Waterloo.
[6] Conneau, A., Schwenk, H., Barrault, L., Lecun, Y., 2016. Very deep convolutional networks for text classification. arXiv preprint arXiv:1606.01781 .
[7] Daudert, T., Ahmadi, S., 2019. Cofif: A corpus of financial reports in french language, in: Proceedings of the First Workshop on Financial Technology and Natural Language Processing, pp. 21–26.
[8] DeRemer, F.L., 1976. Lexical analysis, in: Compiler Construction, Springer. pp. 109–120.
[9] Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 .
[10] Facebook, I., 2016. fastText: Library for fast text representation and classification. URL: https://github.com/facebookresearch/fastText.
[11] Fukushima, K., Miyake, S., 1982. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition, in: Competition and cooperation in neural nets. Springer, pp. 267–285.
[12] Ganesan, K., Zhai, C., 2012. Opinion-based entity ranking. Information retrieval 15, 116–150.
[13] Handler, A., 2014. An empirical study of semantic similarity in wordnet and word2vec .
[14] Kalchbrenner, N., Grefenstette, E., Blunsom, P., 2014. A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 .
[15] Khanuja, S., Dandapat, S., Srinivasan, A., Sitaram, S., Choudhury, M., 2020. Gluecos: An evaluation benchmark for code-switched nlp. arXiv preprint arXiv:2004.12376 .
[16] Kulikowski, C., Ammenwerth, E., Bohne, A., Ganser, K., Haux, R., Knaup, P., Maier, C., Michel, A., Singer, R., Wolff, A., 2002. Medical imaging informatics and medical informatics: Opportunities and constraints. Methods of information in medicine 41, 183–189.
[17] Lamsiyah, S., Mahdaouy, A.E., Ouatik, S.E.A., Espinasse, B., 2021. Unsupervised extractive multi-document summarization method based on transfer learning from bert multi-task fine-tuning. Journal of Information Science , 0165551521990616.
[18] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324.
[19] Leech, G.N., 1992. 100 million words of english: the british national corpus (bnc) .
[20] Lieske, C., McCormick, S., Thurmair, G., 2001. The open lexicon interchange format (olif) comes of age, in: Proceedings of Machine Translation Summit VIII.
[21] Malo, P., Sinha, A., Korhonen, P., Wallenius, J., Takala, P., 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. Journal of the Association for Information Science and Technology 65, 782–796.
[22] Martin, L., Muller, B., Suárez, P.J.O., Dupont, Y., Romary, L., De La Clergerie, É.V., Sagot, B., Seddah, D., 2020. Les modèles de langue contextuels camembert pour le français: impact de la taille et de l'hétérogénéité des données d'entrainement, in: JEP-TALN-RECITAL 2020-33ème Journées d'Études sur la Parole, 27ème Conférence sur le Traitement Automatique des Langues Naturelles, 22ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues, ATALA; AFCP. pp. 54–65.
[23] Martin, L., Muller, B., Suárez, P.J.O., Dupont, Y., Romary, L., de La Clergerie, É.V., Seddah, D., Sagot, B., 2019. Camembert: a tasty french language model. arXiv preprint arXiv:1911.03894 .
[24] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems 26.
[25] Nothman, J., Ringland, N., Radford, W., Murphy, T., Curran, J.R., 2013. Learning multilingual named entity recognition from wikipedia. Artificial Intelligence 194, 151–175.
[26] PaNgB, L., 2005. Exploitingclassrelationshipsforsentimentcate gorizationwithrespectratingsales. IN: ProceedingsofACL r05 .

[27] Patel, K.K., Pal, A., Saurav, K., Jain, P., 2022. Mental health detection using transformer bert, in: Handbook of Research on Lifestyle Sustainability and Management Solutions Using AI, Big Data Analytics, and Visualization. IGI Global, pp. 91–108.
[28] Patient, D., . An unsupervised representation to predict the future of patients from the electronic health records riccardo miotto, li li, brian a. Kidd, Joel T. Dudley Scientific Reports (2016-05-17) https://doi. org/10.1038/srep26094 .
[29] Pennington, J., Socher, R., Manning, C.D., 2014. Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543.
[30] Richman, A.E., Schone, P., 2008. Mining wiki resources for multilingual named entity recognition, in: Proceedings of ACL-08: HLT, pp. 1–9.
[31] Roozen, D., Lelli, F., 2021. Stock values and earnings call transcripts: a dataset suitable for sentiment analysis .
[32] Schuster, M., Paliwal, K.K., 1997. Bidirectional recurrent neural networks. IEEE transactions on Signal Processing 45, 2673–2681.
[33] Simonovsky, M., Komodakis, N., 2017. Dynamic edge-conditioned filters in convolutional neural networks on graphs. arXiv:1704.02901.
[34] Suárez, P.J.O., Sagot, B., Romary, L., 2019. Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures, in: 7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7), Leibniz-Institut für Deutsche Sprache.
[35] Usuga Cadavid, J.P., Grabot, B., Lamouri, S., Pellerin, R., Fortin, A., 2020. Valuing free-form text data from maintenance logs through transfer learning with camembert. Enterprise Information Systems , 1–29.
[36] Wang, H., Li, J., 2022. Unsupervised keyphrase extraction from single document based on bert, in: 2022 International Seminar on Computer Science and Engineering Technology (SCSET), IEEE. pp. 267–270.
[37] Wang, S.I., Manning, C.D., 2012. Baselines and bigrams: Simple, good sentiment and topic classification, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 90–94.
[38] Wang, W.Y., 2017. " liar, liar pants on fire": A new benchmark dataset for fake news detection. arXiv preprint arXiv:1705.00648 .
[39] Wright, R.E., 1995. Logistic regression, in: Reading and understanding multivariate statistics, pp. 217–244.
[40] Xu, B., Shen, H., Cao, Q., Cen, K., Cheng, X., 2020. Graph convolutional networks using heat kernel for semi-supervised learning. arXiv preprint arXiv:2007.16002 .
[41] Yao, L., Mao, C., Luo, Y., 2019. Graph convolutional networks for text classification, in: Proceedings of the AAAI conference on artificial intelligence, pp. 7370–7377.
[42] Yin, W., Kann, K., Yu, M., Schütze, H., 2017. Comparative study of cnn and rnn for natural language processing. arXiv preprint arXiv:1702.01923 .
[43] Zhang, X., Zhao, J., LeCun, Y., 2015. Character-level convolutional networks for text classification. Advances in neural information processing systems 28.