Combining Word Embeddings and Deep Neural Networks for Job Offers and Resumes Classification in IT Recruitment Domain

Amine Habous*[1], El Habib Nfaoui[2]
LISAC Laboratory, Faculty of Sciences Dhar EL Mehraz,
Sidi Mohamed Ben Abdellah University,
Fez, Morocco

***Abstract*—Now-a-days, the use of web portals known as job boards for publishing job offers by recruiters has grown considerably. The candidates in their turn, apply to the job positions via the job boards. Since the opportunities are available on a wide range and the job application process is fast and straightforward, the data flow is transformed to large-volume data sets which are hard to handle. Most companies tend to automate the candidate selection process that aims to match the job offers with suitable resumes. In this paper, we propose a supervised learning approach to classify the job offers and CVs shared in the recruitment sites in order to enhance automatic recruitment process. We used natural language processing techniques for job offers and CV preprocessing. Next, we used word embeddings and deep neural networks to train two models, the first one categorizes recruitment documents based on job skills, and the second one predicts the expertise degree class. The experiment results show that our proposal is very efficient.**

***Keywords*—IT recruitment; word embeddings; deep neural networks; text classification; natural language processing**

I. Introduction

Recruitment is the process of searching and selecting personnel for job positions in the staff of a company [1]. The use of the internet for recruitment purposes has known a prominent increase [2]. The appearance of job boards has made the advertising process easier and more cost-less compared with the old method that uses journal papers and magazines. Furthermore, these job boards provide more advanced services along with publishing job offers for recruiters and candidates curriculum vitae. With the swift development of Information and communication technologies, these portals now support the automatic matching between candidates resumes and their corresponding job offers [3]. The advantages that offer the e-recruitment encourage the companies recruiters and job seekers to upload more data that could be hard to process. Consequently, the data classification task is highly required in order to minimize the effort of handling vast data flows.

The job offer and candidate resume are unstructured textual documents that could have multiple formats. Thus these documents need a delicate process before applying the machine learning algorithms for achieving the classification task [4],[5]. The first phase in this process is the preprocessing task that aims to reduce the multiple forms of words in order to extract relevant features to feed the classification algorithms. Therefore, it reduces the dimensionality of the resulted features, which is an essential part of building an efficient model [6]. The features extraction is a vital part of the whole process, that has a direct impact on the classification model accuracy and performance. The classical model of document representation is the bag of words(BOW), which has many drawbacks regarding the features vector dimensionality and the model accuracy since it ignores the semantics and the contextual relations between the words [7].

Word embeddings presentations carry the semantic meanings beyond the words in textual documents. In 2013, Mikolov et al. proposed the word2vec model [8] that has been widely used in the text processing fields such as sentiment analysis, translation and document summarization, etc. Word2Vec consists on two learning models continuous bag of words (CBOW) and Skip-gram[9]. The CBOW model predicts the current word according to the context words, while Skip gram model predicts the surrounding context words according to the given word.

The IT sector is dynamic and booming in terms of employment, and an essential economic factor, source of national and international development. This sector is growing and evolving rapidly, that's why we have chosen to provide an effective solution to facilitate the processing of 'CV' applications and recruiters job offers. We have decided to address two essential points in the recruitment process, the first being the candidates talent related to the technical skills that they use. The second concerns the degree of expertise related to their experience and other knowledge and academic orientations that distinguish them from others.

In this paper, we propose a solution for text classification using deep neural networks and word embeddings for textual documents in the IT recruitment domain. We aim to classify these documents according to two levels, the first one concerns professional skills and the second is about the degree of expertise. These levels are combined to divide the huge IT jobs databases into smaller subsets to make their processing more easier. For instance, if we have a corpus of 1000000 CVs and 9 classes in the first level and 4 for the second. Thus using our method will allow us to process only 27000 ($\approx$ 1000000/(9$\times$4) considering that the sets are equivalent) to retrieve CVs from the corpus.

The rest of this paper is organized as follows. Section 2 presents the background. Section 3 contains the preliminaries to understand the rest of the paper. Section 4 presents the methodology. Section 5 details and discusses the experiments.

Finally, Section 6 concludes this paper.

II. Background and Related Work

A. Text Classification using Machine Learning

The objective of text classification is to assign a textual document into a set of predefined classes basing on its content. Classification is a supervised learning task where the machine uses labeled data to recognize hidden patterns to classify new documents. Several works have used the traditional techniques of machine learning in text classification combining the SVM (Support Vector Machine) algorithm with the TF-IDF weighting method for documents representation [10],[11] and [12]. Many other works have proposed machine learning based solutions; in [13],[14], the authors used the KNN algorithm and the Naive Bayes probabilistic representation for classifying the text documents. While [15] proposed a technique to enhance the document representation basing on the Naive Bayes method, which estimates the conditional probabilities of Naive Bayes using a deep feature weighting method for text classification.

B. Text Classification using Word Embeddings

Many works have integrated word embeddings based on contextual information for reliable representation instead of using the classical bag of words to classify text documents [16]. The authors in [17] used the deep neural networks methods with pre-trained word embeddings for enhancing Turkish text documents classification. Both of [18] and [19] proposed a solution based on an optimized TF-IDF along with word2Vec for text representation and convolutional neural networks to train the classification model. While [20] use convolutional neural networks on top of pre-trained word vectors for several sentence-level classification tasks. The authors of [21] present an approach that uses topic models based on LDA and word embeddings to represent documents in text categorization problems.

C. Text Classification in the Recruitment Field

Text classification is a vital task in the recruitment process; many techniques have been proposed to precisely classify job seekers resumes, and job offers into the right categories. In [22] paper, the authors presented a machine learning-based solution for binary classification to identify IT job offers on the world wide web. In [23], the authors proposed a deep neural network model for classifying personal basing on competency analysis in the human resources field. The authors in [24] and [25] proposed a text classification method basing on the personality traits in the recruitment domain. [26] proposed a solution for classifying job applicants documents into 27 categories using convolutional neural networks; Since it is hard to obtain resumes, they used free job description snippets for training the model then apply it to the candidates resumes. [27] presented an ontology-based solution for job offers classification, along with [28] that used the same technique for text classification to assist job hazard analysis. In [29], the authors proposed a solution aiming to identify the job skills in a textual document using recurrent neural networks. This method predicts if a word is a skill or not based on its context in the text.

The research works discussed above provide some solutions for textual documents classification in the recruitment process. While they give priority to the relevance of the technological choices and the implementation methods, they lack a business oriented model compared to our solution. Indeed, our proposal is efficient in terms of the chosen technologies and meets the recruitment process requirements.

III. Preliminaries

A. Problem Definition

In this paper, we propose a solution for job offers and candidate CV classification under a set of predefined categories. Let $D = \{x_1, ..., x_n\}$ the set of job offers/CV and $C = \{c_1, ..., c_k\}$ the list of the categories. The classifier estimates probability distribution over categories, and the probability of the correct category should be the highest. The input of the classifier is the training data which is a finite sequence of $\{(x_1, c_1), ..., (x_n, c_k)\}$ pairs from $D \times C$. While the output of the classifier is a function $f : D \to C$ that predicts $c \in C$ for new samples from D.

B. Word Embeddings

The word embeddings presented by [30] is a way to represent the words in textual documents that consist of extracting the contextual relationships between them. This model maps the words or phrases into a low-dimensional continuous space where each dimension represents a specific context. Thus the similar words will have similar vectors. Figure 1 shows an example of the spatial representation of five words according to three contexts.

One of the significant advantages of this model is to deal with dimensionality. As we said before, each word in the text is mapped to a point in a real continuous m-dimensional space that we choose beforehand. Contrary to one-hot encoded feature vector methods such as N-Grams, bag of words, and TF-IDF, the resulted feature vector is enormous. For example, if the corpus size is 100000, and we want to represent a sentence of 5 words, our feature vector will be a 100000-dimensional one-hot encoded vector where only five indexes will have 1.

The words embeddings models can be divided into two categories:

- Count-based methods
- Predictive methods

In count-based models, the semantic similarity between words is obtained by counting the co-occurrence frequency. Technically, we deduce the related words using the co-occurrence matrix. In the predictive models, the word vectors are generated using the predictive ability, i.e., by minimizing the loss between the target word and the context word. Thus the related word vectors are close to each other in space.

C. Deep Neural Networks (DNN)

Generally, the neural network is a technology built to simulate the activity of the human brain in pattern recognition through various layers of simulated neural connections. ‘Deep learning’ is used to describe those networks which represent a specific form of machine learning where technologies use aspects of AI that seek to classify and order information in ways

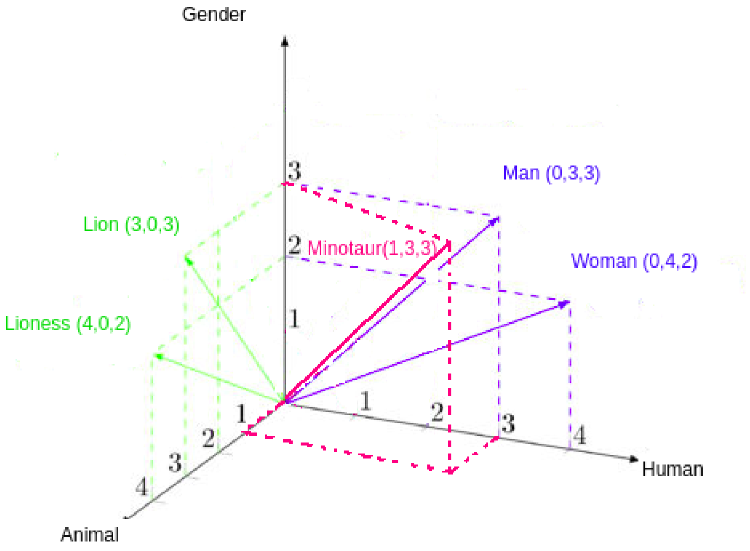


Fig. 1. Special Representation of Words According to Textual Contexts.

that go beyond simple input/output protocols. Mathematically, Deep neural networks are defined as a tuple (L, T, Φ) [31]:

$$L = \{L_k | k \in \{1...K\}\} \tag{1}$$

where L_1 is the input layer, L_K the output one and the other layers are called hidden layers.

$$T \subseteq L \times L \tag{2}$$

is the set of connections between layers

$$\Phi = \{\phi_k | k \in \{2...K\}\} \tag{3}$$

is the set of functions for each layer (non-input layer)

Each layer L_k consists of s_l neurons, each one is associated with two variables: $u_{k,l}$ and $v_{k,l}$ for recording its values before and after an activation function, respectively. There are many activation functions in the literature like the Binary Step function, Linear function, Sigmoid function, Than function and Relu function which is the most popular one for deep neural networks. Except for inputs, every neuron is connected to neurons in the preceding layer by pre-trained parameters such that for all k and l:

$$u_{k,l} = b_{k,l} + \sum_{h=1}^{s_{k-1}} w_{k-1,h,l} \cdot v_{k-1,h} \tag{4}$$

Where :

- $w_{k-1,h,l}$ is the weight of the connection between the h-th neuron of layer k-1 and the l-th neuron of layer k.
- $b_{k,l}$ the bias for neuron $s_{k,l}$
- $v_{k-1,h}$ the result of the activation function for $u_{k-1,l}$

For each input, The DNN assigns a label, that is, the index of the node of the output layer with the largest value $u_{k,l}$. Figure 2 shows the structure of a deep neural network with 4 layers.

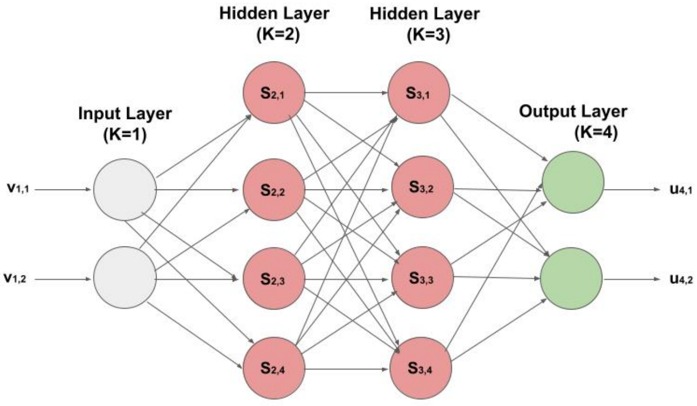


Fig. 2. Example of DNN Composed by 4 Layers.

IV. Methodology

Figure 3 shows the overall proposed architecture. The sub-sections below give all details about each module. In the first sub-section, we describe the labeling process that we use for each model training phase. In further sub-sections, we explain in detail the two models used in the last two sections.

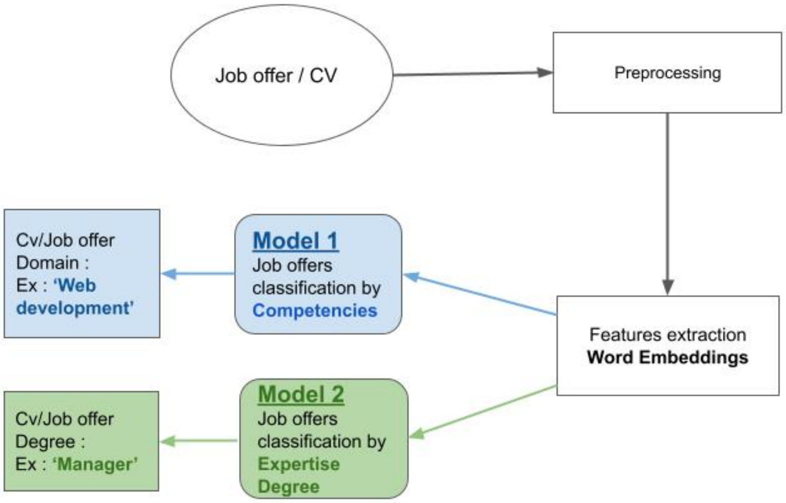


Fig. 3. The Overall Proposed Architecture.

A. The Process of Labeling Job Offers using Job Profiles

In IT recruitment, the job offer is a half structured textual document. It is composed of two parts; the first part is the job header, a set of information about the position, such as the type of contract, the salary, etc. The second part is the job description containing the free text that describes the employer needs, such as job qualifications, technical skills, etc. Most job boards share the same structure as the published job offers. Figure 4 shows an example of a job offer captured from Monster.com

Our work consists of a supervised multi-labeled text classification that takes the training data set with its labels as inputs. Therefore, we utilized the tagging concept using the job profiles to assign the correct labels to text data captured from the job boards. The tagging concept is the process of labeling short textual descriptions or key-words(tags) to information objects [32]. Table I shows some examples of tags that we use for each classification level.

TABLE I. EXAMPLE OF USED TAGS FOR LABELS ASSIGNMENT

	Model 1		Model 2	
Labels	Big data	Web development	engineer	Manager
Tags	Big data - Hadoop Hdfs - Spark	Front end - Back end Full stack Web developper	engineer engineering	Manager Director

We used only the tags that are very representative to avoid any ambiguity in the labeling process. For instance, we used key technologies or IT domains as tags to refer to a category.

B. Preprocessing Step

To preprocess the textual data, we used the traditional method in Natural Language Processing (NLP) [33], [34]:

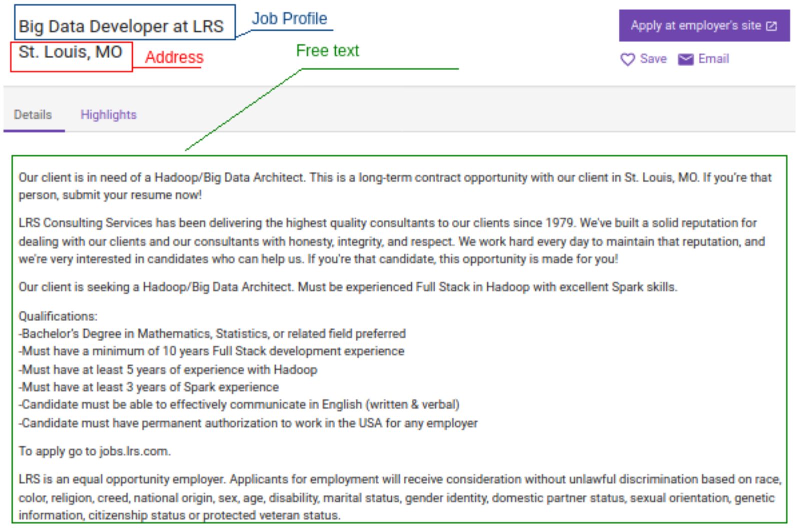
- **Tokenization**: This step aims to transform our textual data into a set of distinct words (tokens)
- **Stop word removal**: After dividing the text into words, we remove the stop words which do not carry any sense using a predefined list of those words.
- **Stemming**: We transform the terms to their stems to unify the document units and reduce the corpus vocabulary size that we will construct in the next step.

These various text preprocessing steps are widely used for dimensionality reduction in the first level. It also helps to enhance the quality of data to promote the extraction of meaningful insights from it.

C. Features Extraction

Generally, the features extraction step in text classification aims to transform a list of words that we extract from the corpus into numerical vectors usable by a classifier. To minimize the vector dimension, we used a predefined list of IT technology fields and their synonyms that we obtained from a domain ontology. Thus we can unify the representation of all the technology fields in the textual corpus. For example, we represent the terms 'BI', 'Business Intelligence' and 'Decision Support System' as the same concept 'BI'; Consequently, it reduces the vector dimension due to the large number of those concepts in the IT recruitment. We can certainly find several open source ontologies such as O*net, ROME, and ESCO. Technically, we used O*net to create a sub-graph of the IT domain fields concepts with their synonyms.

The next step is building the corpus vocabulary, in which every word (token) is indexed based on its frequency in this corpus. Thus we transform each job offer from the corpus to a sequence of integers that correspond to the vocabulary



Big Data Developer at LRS

St. Louis, MO

Our client is in need of a Hadoop/Big Data Architect. This is a long-term contract opportunity with our client in St. Louis, MO. If you're that person, submit your resume now!

LRS Consulting Services has been delivering the highest quality consultants to our clients since 1979. We've built a solid reputation for dealing with our clients and our consultants with honesty, integrity, and respect. We work hard every day to maintain that reputation, and we're very interested in candidates who can help us. If you're that candidate, this opportunity is made for you!

Our client is seeking a Hadoop/Big Data Architect. Must be experienced Full Stack in Hadoop with excellent Spark skills.

Qualifications:
-Bachelor's Degree in Mathematics, Statistics, or related field preferred
-Must have a minimum of 10 years Full Stack development experience
-Must have at least 5 years of experience with Hadoop
-Must have at least 3 years of Spark experience
-Candidate must be able to effectively communicate in English (written & verbal)
-Candidate must have permanent authorization to work in the USA for any employer

To apply go to jobs.lrs.com.

LRS is an equal opportunity employer. Applicants for employment will receive consideration without unlawful discrimination based on race, color, religion, creed, national origin, sex, age, disability, marital status, gender identity, domestic partner status, sexual orientation, genetic information, citizenship status or protected veteran status.

Fig. 4. Example of a Job Offer.

indexes. Since we will use the sequential deep neural networks for training the models, we use sequence padding to ensure that all sequences have the same length. Fig. 5 illustrates the feature extraction process.

D. Model 1: Job Offer Classifications According to Job Competencies

Job competencies are a set of abilities, knowledge, and skills needed to fill a job position or a role within a work environment. In addition to the work experiences that candidates achieve, Those competencies are learned and developed through academic studies. According to IBM corporation, job competencies are categorized as follows:

- Core competencies
- leadership management/business interpersonal
- Job functional competencies
- Job technical competencies
- Technical - task specific

In our context, we will focus on the last two types of competencies. Job technical competencies are the set of knowledge and skills related to the technologies of the IT domain. In contrast, task-specific competencies are the set of missions to perform in a specific role in the IT domain.

As we said before, the published job offers contain a free text part in where the recruiters specify the job requirements needed to fill a given post in his company. We use this data from our corpus to train the first model to classify the new job offers and CVs according to their skills and competencies in the IT domain.

E. Model 2: Job Offer Classifications According to Expertise Degree

In IT recruitment, the degree of expertise of a profile is related to the experience acquired and the employee tasks nature. Thus it is considered as a critical parameter to rank the IT recruitment job offers and candidates. In our paper, we divided these degrees into four separated degrees :

- **Technician**:The technician role is considered as the primary level in IT recruitment. Generally, its academic course does not exceed the Bachelor's degree.
- **Engineer**: The engineering level may require an additional academic degree compared to the technician; in the IT field, an engineer is oriented to be more autonomous in performing tasks. This degree also requires the mastery of many technical and personal skills.
- **Project chief**: This level is more privileged than the others; the project chief manages the cycles of IT projects, so he has many responsibilities requiring more advanced personal skills.
- **Manager/Consultant**: This degree gathers advanced technical, personal, and leadership skills. Managers and Consultants are also experimented with compared to the other levels.

In the IT recruitment field, there are so many ways to classify these roles according to expertise. Thus we focus on finding decisive differences between the chosen ones to train an efficient model for this level.

V. Result and Discussion

In this section, we present the data set that we used to train the proposed models and discuss the experimental results.

A. Data Set

We went through a data capture phase to construct a corpus of job offers and CVs documents from the IT recruitment domain. We have gathered job offers published on the job boards and the recruitment portals using web scraping tools and techniques. We targeted job offers for two reasons :

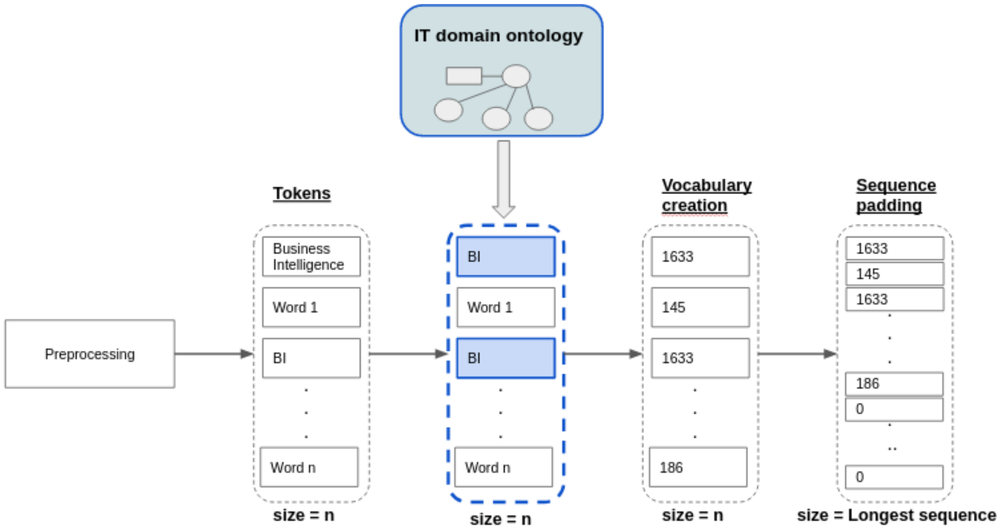


Fig. 5. Features Extraction Steps.

- The job offers are widely available and free compared to candidate CVs which are rarely published and shared on job boards; Consequently, it is challenging to create a CV corpus.
- The job offers are published according to a job profile that we use to define the class for supervised learning; whereas, the CVs contain a set of job profiles that the candidate achieved in his career.

We gathered about 10 000 documents from E-recruitment sites that share french job offers. Therefore, our tests concern French language. We preprocessed those documents following the steps explained in section IV.B above.

In the next step, we focused on filtering the job offers by removing the redundant ones. Then we filter the job offers having job profiles that don't contain the tags which makes the process of labeling more difficult. Table II and Table III describe the data that we used to train the two models.

TABLE II. The Description of the Data Set used to Train Model 1

Class	Number of job offers
Web/software development	1115
Network engineering	1061
Embedded software engineering	796
Testing engineering	758
Business intelligence	740
Big data development	635
Data science	396
Information systems management	307
Database administration	245
Total	6053

To evaluate the two models performance, we split the data set for each of them as follows:

- Train set : 60 %
- Validation set : 20 %
- Test set : 20 %

TABLE III. The Description of the Data Set used to Train Model 2

Class	Number of job offers
Technician	437
Engineer	4486
Project chief	1635
Consultant/Manager	1554
Total	8112

B. Experimental Results

To implement the solution, we used the Python language for implementing and evaluating our algorithms on an HP 4540s computer with an Intel i5 CPU running at 3.20 GHz with Ubuntu 18.04 (64-bit) and 8GB of RAM.
We used the NLTK library for the NLP tasks and the OWL-Ready2 to browse the IT ontology in the features extraction step. We trained the deep learning models using the KERAS toolkit. Our DNN models are composed of an embedding layer in the input. Then, two hidden layers with dropout ones to avoid overfitting.

The Table IV shows the two models performance using the accuracy, recall and f-score measures.

TABLE IV. Model 1 and 2 Performance Results

	Model 1	Model 2
Accuracy	90.41 %	82.93 %
Recall	85.85 %	75.10 %
F-Score	88.07 %	78.82 %

The confusion matrices for job offers classification are shown in Fig. 6 and Fig. 7 for model 1 and 2 (resp.)

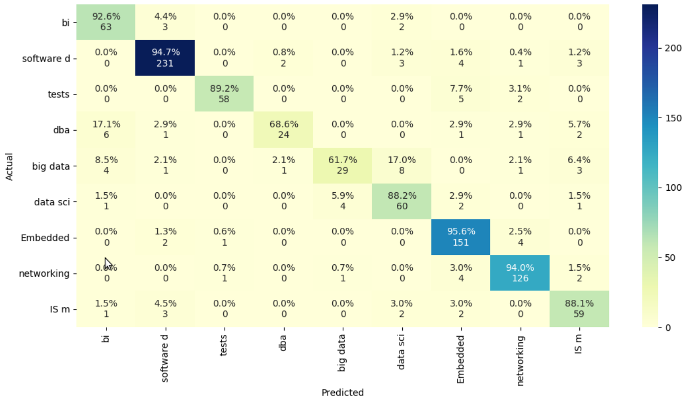


Fig. 6. Resulted Confusion Matrix for Model 1.

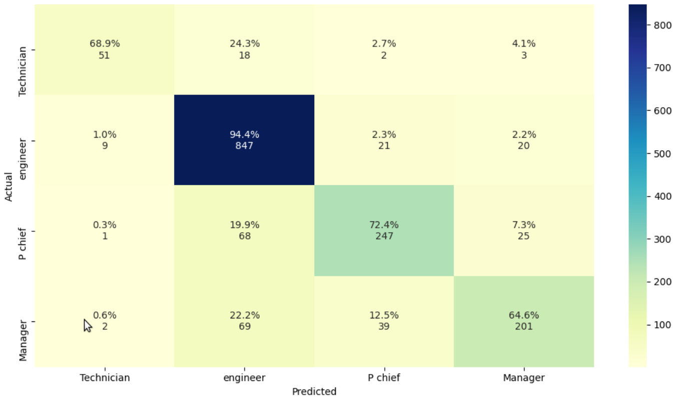


Fig. 7. Resulted Confusion Matrix for Model 2.

The results of the job offers classification show a good performance regarding the two models. They prove the relevance of the optimal choices that we made. The confusion matrix from figure 6 tells us that the minimum precision value for model 1 is 61 % for the "big data" class, which is justified by the complexity of the IT recruitment field. For example, in some cases, we can confuse a "data scientist" and a "big data developer" given a large number of technical skills in common. Concerning model 2, the confusion matrix from Fig. 7 shows us a low precision for the "manager" degree, which is predictable given few the job offers used for training the models.

In the implementation phase, we have tried multiple methods for textual classification in the recruitment process. First, we have trained a word2Vec model to catch similarities for the job offers corpus. We have also used the pre-trained model Glove for word embeddings. In both cases, the first results obtained with the Keras embedding layer outperform the other methods. The specific context of IT recruitment helps in semantic relationships capture with the deep neural network embedding layer. However, to train an embedding model, a large volume of data is necessary, which is not available in our case. In general, the pre-trained Glove model works well for text classification, but our solution concerns a specific context of words.

VI. Conclusion

In this paper, we proposed a technique for CV and job offers classification to facilitate their processing in massive data flows. We used the deep learning networks combined with word embeddings to build two efficient models. The first one aims to categorize CV and job offers according to the technical competencies in the IT domain, while the second one classifies these documents basing on the required or the recommended expertise degree.

The tests have shown that the constructed models give efficient results in terms of classification measures. We have well described the data set used in the results section, to show the efficiency of the models in such a complex context.

For future work, we plan to increase the classification classes number to have more subsets of the massive data; therefore, we gain additional performance in their processing. In addition, we aim to apply this solution to several other business sectors other than the IT field.

References

[1] V. Lytvyn, V. Vysotska, P. Petro, g. Bobyk, and B. Pakholok, "A method for constructing recruitment rules based on the analysis of a specialist's competences," *Eastern-European Journal of Enterprise Technologies*, vol. 6, pp. 4–14, 12 2016.

[2] J. Chen, Z. Niu, and H. Fu, "A novel knowledge extraction framework for resumes based on text classifier," in *Web-Age Information Management* (X. L. Dong, X. Yu, J. Li, and Y. Sun, eds.), (Cham), pp. 540–543, Springer International Publishing, 2015.

[3] A. Kmail, M. Maree, M. Belkhatir, and S. Alhashmi, "An automatic online recruitment system based on exploiting multiple semantic resources and concept-relatedness measures," 11 2015.

[4] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[5] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, A. Mehmood, and M. T. Sadiq, "Document-level text classification using single-layer multisize filters convolutional neural network," *IEEE Access*, vol. 8, pp. 42689–42707, 2020.

[6] A. I. Kadhim, "An evaluation of preprocessing techniques for text classification," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 16, no. 6, 2018.

[7] B. Heap, M. Bain, W. Wobcke, A. Krzywicki, and S. Schmeidl, "Word vector enrichment of low frequency words in the bag-of-words model for short text multi-class classification problems," *arXiv preprint arXiv:1709.05778*, 2017.

[8] X. Rong, "word2vec parameter learning explained," *CoRR*, vol. abs/1411.2738, 2014.

[9] L. Ma and Y. Zhang, "Using word2vec to process big text data," in *2015 IEEE International Conference on Big Data (Big Data)*, pp. 2895–2897, 2015.

[10] G. Forman, "Bns feature scaling: An improved representation over tf-idf for svm text classification," pp. 263–270, 01 2008.

[11] K. Chen, Z. Zhang, J. Long, and H. Zhang, "Turning from tf-idf to tf-igm for term weighting in text classification," *Expert Systems with Applications*, vol. 66, pp. 245 – 260, 2016.

[12] F. Luo, C. Li, and Z. Cao, "Affective-feature-based sentiment analysis using svm classifier," in *2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 276–281, 2016.

[13] V. Bijalwan, V. Kumar, P. Kumari, and J. Pascual, "Knn based machine learning approach for text and document mining," *International Journal of Database Theory and Application*, vol. 7, no. 1, pp. 61–70, 2014.

[14] P. Bolaj and S. Govilkar, "Text classification for marathi documents using supervised learning methods," *International Journal of Computer Applications*, vol. 155, no. 8, pp. 6–10, 2016.

[15] L. Jiang, C. Li, S. Wang, and L. Zhang, "Deep feature weighting for naive bayes and its application to text classification," *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 26–39, 2016.

[16] A. El Mahdaouy, E. Gaussier, and S. O. El Alaoui, "Arabic text classification based on word and document embeddings," in *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016* (A. E. Hassanien, K. Shaalan, T. Gaber, A. T. Azar, and M. F. Tolba, eds.), (Cham), pp. 32–41, Springer International Publishing, 2017.

[17] M. Aydoğan and A. Karci, "Improving the accuracy using pre-trained word embeddings on deep neural networks for turkish text classification," *Physica A: Statistical Mechanics and its Applications*, vol. 541, p. 123288, 2020.

[18] L. Li, L. Xiao, W. Jin, H. Zhu, and G. Yang, "Text classification based on word2vec and convolutional neural network," in *Neural Information Processing* (L. Cheng, A. C. S. Leung, and S. Ozawa, eds.), (Cham), pp. 450–460, Springer International Publishing, 2018.

[19] T. Zhang and L. Wang, "Research on text classification method based on word2vec and improved tf-idf," in *Advances in Intelligent Systems and Interactive Applications* (F. Xhafa, S. Patnaik, and M. Tavana, eds.), (Cham), pp. 199–205, Springer International Publishing, 2020.

[20] A. Mandelbaum and A. Shalev, "Word embeddings and their use in sentence classification tasks," 10 2016.

[21] J. M. Lima and J. Maia, "A topical word embeddings for text classification," pp. 25–35, 10 2018.

[22] E. ul haq Dar and J. Dorn, "Classification of job offers of the world wide web," in *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1–8, 2018.

[23] Q. Guohao, W. Bin, W. Bai, and Z. Baoli, "Competency analysis in human resources using text classification based on deep neural network," in *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)*, pp. 322–329, IEEE, 2019.

[24] E. Faliagka, A. Tsakalidis, and G. Tzimas, "An integrated e-recruitment system for automated personality mining and applicant ranking," *Internet research*, 2012.

[25] S. Bharadwaj, S. Sridhar, R. Choudhary, and R. Srinath, "Persona traits identification based on myers-briggs type indicator(mbti) - a text classification approach," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1076–1082, 2018.

[26] L. Sayfullina, E. Malmi, Y. Liao, and A. Jung, "Domain adaptation for resume classification using convolutional neural networks," in *Analysis of Images, Social Networks and Texts* (W. M. van der Aalst, D. I. Ignatov, M. Khachay, S. O. Kuznetsov, V. Lempitsky, I. A. Lomazova, N. Loukachevitch, A. Napoli, A. Panchenko, P. M. Pardalos, A. V. Savchenko, and S. Wasserman, eds.), (Cham), pp. 82–93, Springer International Publishing, 2018.

[27] E. ul haq Dar and J. Dorn, "Ontology based classification system for online job offers," in *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1–8, 2018.

[28] N.-W. Chi, K.-Y. Lin, and S.-H. Hsieh, "Using ontology-based text classification to assist job hazard analysis," *Advanced Engineering Informatics*, vol. 28, no. 4, pp. 381 – 394, 2014.

[29] D. Baad *et al.*, "Automatic job skill taxonomy generation for recruitment systems," 2019.

[30] G. E. Hinton *et al.*, "Learning distributed representations of concepts," in *Proceedings of the eighth annual conference of the cognitive science society*, vol. 1, p. 12, Amherst, MA, 1986.

[31] Y. Sun, X. Huang, and D. Kroening, "Testing deep neural networks," *CoRR*, vol. abs/1803.04792, 2018.

[32] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Multilabel text classification for automated tag suggestion," in *Proceedings of the ECML/PKDD*, vol. 18, p. 5, 2008.

[33] V. Gurusamy and S. Kannan, "Preprocessing techniques for text mining," 10 2014.

[34] S. Vijayarani, M. J. Ilamathi, and M. Nithya, "Preprocessing techniques for text mining-an overview," *International Journal of Computer Science & Communication Networks*, vol. 5, no. 1, pp. 7–16, 2015.