A Two-Phase Framework for Automated Information Extraction from Curriculum Vitae

Tan Minh Ngo
IBM Vietnam
minh.tan@ibm.com

Quynh Ngoc Nguyen, Thuc Duy Nguyen
International School, Vietnam National University, Hanoi
{quynhnn, thucnd}@vnuis.vn

Oanh Thi Tran[1]✉
International School, Vietnam National University, Hanoi
oanhtt@gmail.com

2023 RIVF International Conference on Computing and Communication Technologies (RIVF) | 979-8-3503-1584-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/RIVF60135.2023.10471865

***Abstract*— Information extraction from CV is the task of extracting relevant information from CVs without human intervention by using NLP and ML techniques. To solve the task, we propose an end-to-end framework that consists of two main phases, namely Block segmentation and NE recognition. In the first step, we consider CVs as images and then exploit the image segmentation techniques to partition images into non-overlapping information regions such as Personal Information, Educational Background, Working Experience regions, etc. In the following step, the relevant information in each region is detected using its contextual information. In this second step, different robust deep learning techniques are investigated. In addition, to facilitate conducting experiments, we also introduce a dataset which includes 914 English CVs and 400 Vietnamese CVs. These CVs were collected, pre-processed and manually annotated with rich information such as names, phone numbers, email addresses, organizations, job positions, degrees, and skill sets. We conduct extensive experiments and achieve about 80% in the F1 score on this benchmark dataset using the XLM-Roberta model. This result is very promising for real-world applications and will be exploited in a variety of ways.**

Keywords—information extraction, CV, resumes, pre-trained models, BERT

I. INTRODUCTION

Job boards offer companies access to a large pool of potential job seekers, making efficient talent acquisition processes crucial for business success. However, talent acquisition departments face challenges in identifying suitable candidates due to the time-consuming task of reviewing numerous CVs across various platforms. This led to the rise of a typical task, namely automated information extraction from CVs. The goal is to efficiently analyze a large number of CVs and automatically extract key details such as candidate names, contact information, education history, work experience, skills, and certifications.

By employing machine learning techniques, the extraction system can understand the structure and content of CVs. Over the years, various approaches have been explored to tackle this task, leveraging advancements in natural language processing and machine learning. In the early research stages, researchers used rule-based systems to extract information from CVs [10]. These systems relied on predefined patterns and heuristics to identify and extract specific data fields, such as names, contact details, and work experience. Unfortunately, these systems were limited in flexibility and struggled with the diversity of CV formats. Recently, when machine learning and deep learning advances, many supervised learning models are investigated such as decision tree and logistic regression [11], neural network-based classifiers and distributed embeddings [1], Document object model tree structure [2], BERT language model [8].

Most of these researches were dedicated to popular languages, especially English, not much work was done on poorly-resourced languages like Vietnamese. To our knowledge, there existed only one work of Nguyen et al., [18] which proposed a solution consisting of four phases. They combined CNN, biLSTM and CRF to extract information from CVs. In this paper, motivated by the pre-trained models and existing researches, we propose an end-to-end framework which includes two main phases, namely Block segmentation and NE recognition. In the first step, we consider CVs as images and then exploit the image segmentation techniques to partition images into non-overlapping regions such as Personal Information region, Education region, Working Experience region, etc. Next, we extract the relevant information in each region using its contextual information and store it in a database for later use. In this second step, different robust deep learning techniques such as BERT, XLM-Roberta are investigated. In addition, to facilitate conducting experiments, we also introduce a dataset which includes 914 English CVs and 400 Vietnamese CVs. Based on this dataset, we performed extensive experiments to make comparison and show the effectiveness of the proposed framework. We achieve 80% in the F1 score on this benchmark dataset using the XLM-Roberta model.

The remainder of this paper is organized as follows: Section 2 shows related work on extracting information from CVs. Section 3 presents our proposed 2-phase framework. In Section 4, experimental setup, experimental results and model deployments are shown. Finally, we conclude the paper and point out some future lines of work in Section 5.

II. RELATED WORK

CV information extraction, also called CV parsing helps extracting relevant information from CVs. This allows us to convert from unstructured data into structured data. There are two main approaches which are rule-based and machine learning-based ones. The former involves the utilization of keywords or texts obtained from various sources retrieval to set up rules [19]. However, these algorithms or processes have

[1] Corresponding author

979-8-3503-1584-4/23/$31.00 ©2023 IEEE

limitations, such as low precision and unacceptable ranking outcomes due to the introduction of noise.

To improve the accuracy of the extraction models, researchers have proposed novel methods which exploit both traditional machine learning and recent deep learning innovation. For instance, in [11] the authors used Optical Character Recognition (OCR) to extract the data from Resume. Then, the techniques in Natural Language Processing and Ranking Algorithm are exploited to rank the resume according to the particular companies. In [1], the authors suggested using a BiLSTM-CNNs-CRF and text block segmentation techniques to extract essential features for recognizing named entity recognition within labeled text blocks. In [6], authors presented some methods to extract information from CVs and then make recommendations by using different traditional machine learning methods such as k-NN, SVM and Naive Bayes. In [7], authors used DT and LR to recognize important information and then rank it according to the preference of the associated company and requirements. In [8], authors employs the transformer architecture and its multilingual implementation of the encoder part (i.e. BERT) to extract useful information from multilingual, unstructured CV documents. [20] combines Neural Network and Conditional Random Fields for Efficient Resume Parsing.

We have witnessed that a lot of work on extracting information from CVs has been done for popular languages, especially English. Unfortunately, not much work has been done for poorly-resourced languages like Vietnamese. In this work, motivated by the pre-trained models and existing researches, we propose a framework for CV information extraction by combining the ideas from block segmentation, identification, and NER models to solve the task. We also contribute a benchmark dataset in both English and Vietnamese to facilitate conducting comparison experiments.

III. A PROPOSED TWO-PHASE FRAMEWORK TO EXTRACT INFORMATION FROM CVS

The proposed framework is illustrated in Figure 1. It includes the two main following phases:

- **CV Block Segmentation**: This phase divides a CV into non-overlapping regions to make the subsequent information extraction phase easier. Each region contains a distinct type of information about a candidate. For example, the personal information region commonly contains the information about the name, the phone number, the email addresses, etc. of each candidate.

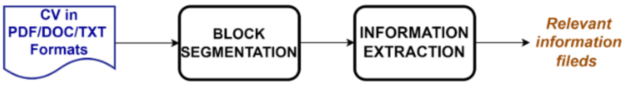


FIG. 1. AN END-TO-END PROPOSED FRAMEWORK FOR CV INFORMATION EXTRACTION

- **Information extraction model**: Recognize the corresponding information in each region. In this paper, there are seven main relevant information fields of our interests which are name, phone number, email address, organization, job position, degree, and skill sets. These information fields are saved as structured data in a database and used in a variety of ways.

A. CV Block segmentation by Layout Analysis (LXML)

In this section, instead of using the traditional approach of using common headings, we propose segmentation using the layout analysis approach. This proposed method can solve the disadvantages of the traditional approach such as handling misspelled headings, duplicate appearance, and utilizing the original structure of CVs.

Every CV has multiple blocks which have a similar structure but provide different information fields. Figure 2 shows an example. Each block starts with a heading followed by its related information. By identifying and extracting information in these blocks separately, the accuracy of the output will be significantly increased. We focus on four important segments which are Personal information, Education, Work Experiences, and Skills.

This method identifies blocks of words/sentences through the coordinate in the CV files. Then it determines which heading and information block are related to each other.

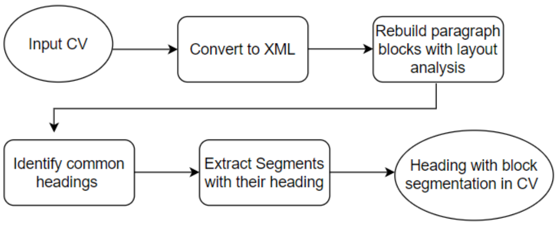

Input CV
Convert to XML
Rebuild paragraph blocks with layout analysis
Identify common headings
Extract Segments with their heading
Heading with block segmentation in CV

FIG. 2. LAYOUT ANALYSIS WORKFLOW DIAGRAM

1) Converts PDF/TXT/DOCX to XML.

The process skips text boxes and lines information, keeping only word information including coordinate, color, font size, and font family. Then words are analyzed through a layout analysis module to rebuild paragraphs and update the paragraph information in the XML object.

2) Rebuild paragraph blocks with layout analysis.

The next step is to put the paragraphs into a sentence segmentation module - a deep learning model to rebuild sentences. This step solves the problem where sentences have no period symbol at the end. Information about sentences are updated in the XML object. As a result, the next step uses a split and merge algorithm to solve the problem as shown in Figure 3.

Split (Line -> Lines): Use K Means algorithm based on the horizontal distance between words with 2 cluster:

- Distance between words in the same line.
- Distance between words in different lines.

Merge (Lines -> Blocks): Use Hierarchical clustering (HAC) algorithm to merge closed lines to each other. HAC is used with single linkage criteria (distance between two clusters equal to the minimum distance of two members of two clusters. Threshold value equals to 3 (stop when distance of two lines is equal to 3 times the height of those lines). Raw text blocks will be identified. However, paragraphs inside those blocks need to be split.

Split (Block -> Blocks): Extract each paragraph into single blocks. K-Means algorithm is exploited using the vertical distance between lines with 2 clusters:

- Distance between lines in the same paragraph.
- Distance between lines in different paragraphs.

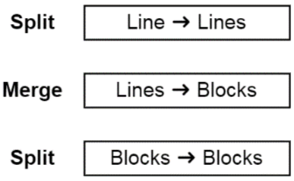


FIG. 3.SPLIT AND MERGE ALGORITHM

3) *Identify common headings*

This paper is interested in four common segments. In this step, we will build a dictionary file which contains common heading titles for each segment. For example, the education segment maybe started by the keywords like Educational Background, Education, Educational Qualification, Education Section, etc. This dictionary will be exploited in the subsequent step in order to extract the correct segment.

4) *Extract segments with their headings*

Identified at least one heading with the common heading method above. However, the Levenshtein distance algorithm is also applied to improve misspell problems. Headings usually have some special characteristics compared to normal words such as: larger font size, position is closer to the left of the CV, different font family, and contain uppercase letters, etc. Those characteristics of the found headings are used to search for the others. The final step is to get related paragraphs with the information we build in the XML object.

B. *Information extraction model*

In this section, recognizing key information fields is considered as a sequence labelling problem. Each word is assigned one label which indicates whether it is the beginning (B), the middle (I) or outside (O) the boundary of any entities using IOB notation. To this end, we exploited deep learning approaches. Figure 4 shows the general architecture of the learning model. It includes three main parts as follows:

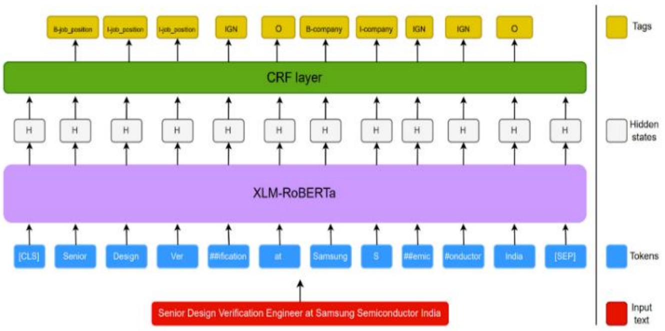


FIG. 4: INFORMATION EXTRACTION ARCHITECTURE

- ***Tokenizing Layer***: The utterance is split into different tokens. Each input text is split into sub-words unit using a WordPiece tokenizer. Then, each token is initialized and trained jointly with the rest of the model.
- ***Text Learning Representation***: In this layer, we exploit large language models (LLMs), specifically XLM-Roberta to learn the representation based on token embeddings. A LLM is a pre-trained deep-learning model that can comprehend and produce text in a manner like that of a human.
- ***Tag/Label Inference Layer***: It takes the output of the previous layer and then applies weights to give the final probabilities for each label using CRFs. CRFs are better at modeling the relation between neighboring labels in this sequence labeling task.

IV. EXPERIMENTS

A. *Dataset annotation*

Figure 5 shows the dataset annotation steps. It includes three main steps which are (1) collecting raw data, (2) pre-processing data, and (3) annotating labels using INCEpTION tool.

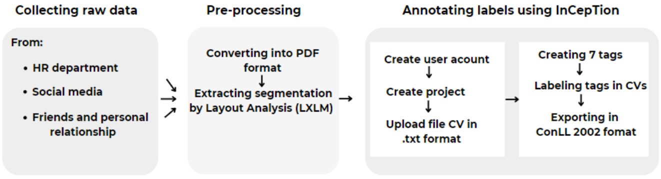

Collecting raw data
From:
• HR department
• Social media
• Friends and personal relationship
Pre-processing
Converting into PDF format
Extracting segmentation by Layout Analysis (LXLM)
Annotating labels using InCepTion
Create user acount
Create project
Upload file CV in .txt format
Creating 7 tags
Labeling tags in CVs
Exporting in ConLL 2002 fomat

FIG. 5: DATA ANNOTATION PROCESS

1) *Collecting raw texts*

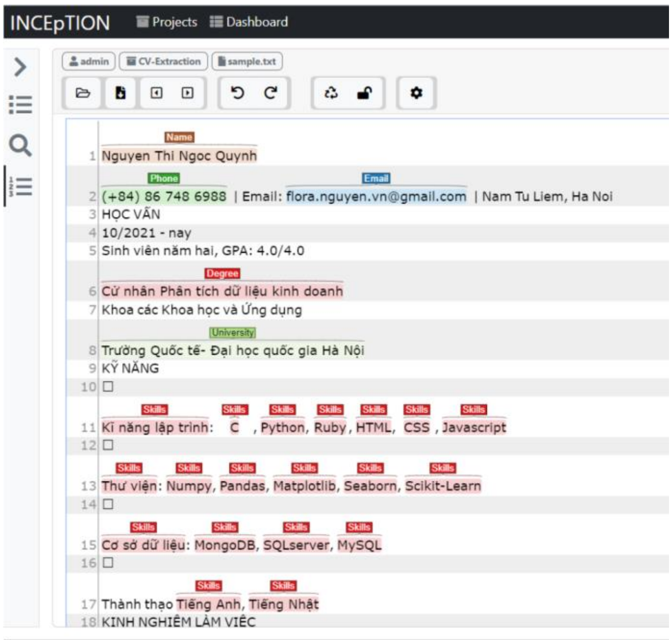
Our data collection takes place over 3 months, across a variety of platforms and methods such as: (1) From cooperation with some recruitment departments of technology companies, (2) Make a call for help with data from individuals on social media, and (3) Collected directly from friends or other personal relationships.

2) *Data preprocessing*

This process involves two steps. Initially, CVs with different formats are changed into PDF and merged into one file for faster processing in the next step. Then, they experience segmentation by Layout Analysis method to extract all texts from PDF into txt file for annotating dataset because annotation in PDF format may cause errors in the training dataset.

3) *Annotating Dataset*

After carefully researching and testing, we came up with a solution for labeling data, which could easily be one of the most critical aspects of the project because this process has the biggest contribution to the final result of extracting information from CVs. The tool we decided to use is INCEpTION and we deployed it on OpenShift. For further insights, INCEpTION is a platform for semantic annotations that provides knowledge management and intelligent help. INCEpTION allows several users to collaborate on the same project simultaneously and may hold multiple projects simultaneously.



Nguyen Thi Ngoc Quynh
(+84) 86 748 6988 | Email: flora.nguyen.vn@gmail.com | Nam Tu Liem, Ha Noi
HỌC VẤN
10/2021 - nay
Sinh viên năm hai, GPA: 4.0/4.0
Cử nhân Phân tích dữ liệu kinh doanh
Khoa các Khoa học và Ứng dụng
Trường Quốc tế- Đại học quốc gia Hà Nội
KỸ NĂNG
□
Kĩ năng lập trình: C , Python, Ruby, HTML, CSS , Javascript
□
Thư viện: Numpy, Pandas, Matplotlib, Seaborn, Scikit-Learn
□
Cơ sở dữ liệu: MongoDB, SQLserver, MySQL
□
Thành thạo Tiếng Anh, Tiếng Nhật
KINH NGHIÊM LÀM VIÊC

FIG. 6. LABELING DEMONSTRATION PROCESS

Moving on to the actual work, first, we created 7 tags including name, phone number, email, organization name, degree name, job position name, and skills for annotating resumes from the dataset. Then, we labeled the CVs in the text file (.txt files) by highlighting important attributes in a CV. For example in Figure 6, we highlight the entity's name, email, and phone number. Finally, we exported labeled data using ConLL 2002 format for further processing.

After successfully labeling all the data, we had to check for mutual agreement between annotators. Our group decided to calculate this based on Kohen's Kappa. For subjective (all-out) things, Cohen's kappa coefficient is a measurement that is utilized to survey between rater dependability as well as interrater unwavering quality. We measured this coefficient and received a result of 0.852, which stands for almost perfect annotation between annotators. Some statistics about the final dataset are shown in Figure 7.

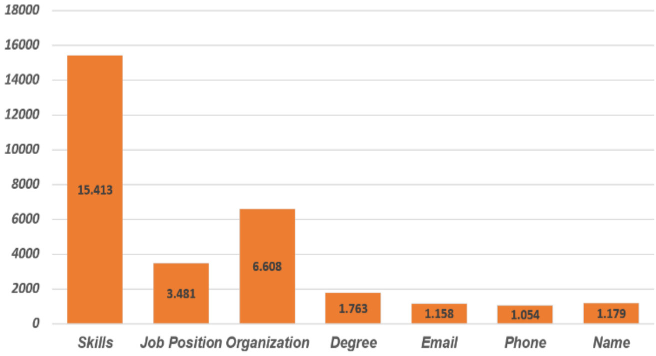


FIG. 7. LABELS STATISTICS

B. Experimental Setups

The dataset is merged into the mixed data. The data file will be trained by several robust deep learning models. We conducted experiments on three different sequence labelling models. The data format we used to train models is ConLL 2002, the data is split into training, test, and validation datasets with a ratio of 70/20/10. We used training set to train the model, development set to fine-tune the parameters of the models, and the test set to measure the performance of the model. Model evaluation is conducted by employing several evaluation measures which are precision, recall and F1 scores. In the early stages of research, it is crucial to evaluate a model's effectiveness. The model evaluation also aids in model monitoring. All models are run and tested on the Google Colab platform.

C. Experimental Results

In this section, we present two kinds of experiments. First, we show the results of the block segmentation part. Then, the results of NER models are shown for each language mentioned.

1) CV block segmentation method using LXML

Experiments were conducted to compare the proposed method with the conventional approach which segments blocks by using common heading rules. This baseline method performs segmentation by identifying common headings then extracts the region attached with its related information. All CVs are converted to text by the available converting tools before being processed.

TABLE 1. SEGMENTATION COMPARISON RESULT

Regions	Segment by common headings	Segment by LXML (Layout Analysis)
Personal Info	90.14%	99.42%
Education	85.22%	98.15%
Work experience	80.43%	95.66%
Skill	80.16%	93.93%

Each of the four segments has data-dictionary files which store common heading names. All of the text between the current found heading and the next found heading is recognized as a part of the segment. Since the research here focuses on only four segments, headings of other segments are also considered to be the end of those four. Personal information is a special case when this segment always appears at the start of the CV. Therefore, all of the texts from the start to the first found heading is considered to be in the personal information segment. The information of this segment can also be added by finding email, phone number, etc using the corresponding regular expressions.

With more complex algorithms added to handle raw input, the output of the proposed method is obviously better (as shown in Table 1). Drawbacks of the former segmentation method are well-handled in the latter method as follows:
- *Misspelled headings*: with the Levenshtein distance algorithm, misspelled headings can still be recognized.
- *Common heading file cannot cover all cases*: The first method requires searching headings of all segments with common heading files. While the latter requires only one of them to find the rest of the heading.
- *Duplicate appearance*: The second method does not loop to all words of the CV to find headings. Therefore, duplicate appearance in the body of the paragraph makes no problems.
- *Cannot keep PDF structure*: Layout analysis handle input and keep original
structure.

2) Experimental results of NER models

We have conducted experiments on three different pre-trained language models which are *xlm-roberta-base, bert-base-multilingual-cased, bert-base-uncased*. The experimental results are showned in Table 2.

Models	**Precision**	**Recall**	**F1**
xlm-roberta-base	73.36	**87.17**	**79.67**
bert-base-multilingual-cased	**76.41**	80.27	78.29
bert-base-uncased	72.83	84.05	78.04

TABLE 2. RESULTS COMPARISON BETWEEN MODELS

The evaluation results of our study indicate that all three models, namely xlm-roberta-base, bert-base-multilingual-cased, and bert-base-uncased, performed well on information extraction tasks. Among those models, *xlm-roberta-base* outperformed the other two models in terms of Recall and F1-score, with an Precision of 87% and an F1-score of 80%. Specifically, *xlm-roberta-base* achieved the highest precision, recall, and F1-score in all categories of named entities, including personal name, organization, degree, email, phone, job position and skills. The model also showed a higher ability to recognize rare and out-of-vocabulary named entities compared to the other two models. It is worth noting that although *bert-base-multilingual-cased* and *bert-base-uncased* models also showed promising results, their performance was slightly lower compared to *xlm-roberta-base*. We believe that this is due to the fact that *xlm-roberta-base* has been pre-trained on a larger and more diverse set of languages, which allowed it to capture more complex linguistic patterns and context.

Overall, based on our evaluation results, we recommend the use of *xlm-roberta-base* for information extraction tasks, especially in multilingual contexts where the recognition of named entities in different languages is required. By applying this model in Information Extraction tasks for Resume/CV, our tools will help the talent acquisition departments and businesses improve their outputs, thereby saving their time and resources in recruitment processes, and data management.

We also evaluate the Precision, Recall and F1-Score of different entities individually to identify which type of entity is more difficult for the model to identify. Table 3 indicated that for *degree*, *organization* and *job position* entity, the F1-score was fairly lower than *name, phone* and *email* entity. It is because these three entities contain more words in a chunk compared to other entities. For example, with the entity *degree,* the length could reach to 10 words such as "Bachelor of Science in Civil and Environmental Engineering (Honors Degree)". These entities are without context, so the model mainly uses mnemonic ability to predict. It is this point that makes the generality of the model not high and the evaluation results on the test data set much lower than other types of information. Therefore, we can conclude that the identifying *degree*, *organization* and *job position* entities are much more challenging to other type of named entities.

Entity	**Precision**	**Recall**	**F1**
Name	90.12	88.78	89.44
Phone	92.91	92.87	92.89
Email	92.11	92.89	92.50
Degree	74.34	81.89	77.93
Organization	72.11	73.02	72.56
Job position	70.65	80.01	75.04
Skills	76.23	82.10	79.06

TABLE 3. RESULTS ON EACH ENTITY TYPES OF THE BEST MODEL

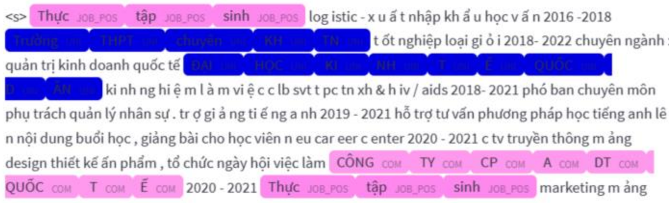
D. Model Deployment

We use streamlit to deploy the model on the webpage platform. To be more specified, the web apps structure should contain three main "*directories*", including:
- Folder: resource – store all of the material such as labels file, pre-trained model in use, picture, …
- Folder: resumes – store our resumes to be extracted in the features
- Last part: python web apps file (streamlit) – store all the code inside to load the model, cv files, or run and return prediction.

In advance, it is crucial to have the readme file (markdown type) to note for all the users about the project description, installation, and so on, and the requirements.txt to list all packages used in the current project .

The deployment process, in general, will consist of three phases:

- Firstly, loading models is crucial as we will use our outcome as a pre-trained model for the background running process. In our project, we were using the xlm-roberta-base model and saved the full model with a .pt file.



<s> Thực JOB_POS tập JOB_POS sinh JOB_POS log istic - x u ấ t nhập kh ẩ u học v ấ n 2016 -2018 Trường UNI THPT UNI chuyên UNI KH UNI TN UNI t ốt nghiệp loại gi ỏ i 2018- 2022 chuyên ngành : quản trị kinh doanh quốc tế ĐẠI UNI HỌC UNI KI UNI NH UNI T UNI Ế UNI QUỐC UNI D UNI ÂN UNI ki nh ng hi ệ m l à m vi ệ c c lb svt t pc tn xh & h iv / aids 2018- 2021 phó ban chuyên môn phụ trách quản lý nhân sự . tr ợ gi ả ng ti ế ng a nh 2019 - 2021 hỗ trợ tư vấn phương pháp học tiếng anh lê n nội dung buổi học , giảng bài cho học viên n eu car eer c enter 2020 - 2021 c tv truyền thông m ảng design thiết kế ấn phẩm , tổ chức ngày hội việc làm CÔNG COM TY COM CP COM A COM DT COM QUỐC COM T COM Ế COM 2020 - 2021 Thực JOB_POS tập JOB_POS sinh JOB_POS marketing m ảng

Fig 8 – An example of Streamlit deployment

- Secondly, we will focus on creating the make_prediction function. In this function, we will include the model in the running process and initialize tokens and ways to split them. While implementing the code, it is crucial to research about how the splitting process worked.This function will then return the result and be added to the next phase.

- Lastly, this is our main part to build up a load_file function. With Streamlit, we can easily upload files using the built-in function st.file_uploader and also customize the upload types which are pdf, docx, and text plain and return the annotated text with a specified label and color.

V. CONCLUSION

This paper introduced an end-to-end framework to obtain beneficial data from CVs which may exist in a variety of common file types such as DOCX, PDF, or TXT. This framework consists of two main phases, namely CV block segmentation and NER models. It helps to eliminate the need for manually developing various hand-crafted features, which can be a time-consuming task. Our approach achieved an 97% accuracy in CV Section Segmentation using KMeans and HAC techniques, which we found to be satisfactory. After extensive conducting experiments on three models, we determined that *XLM-RoBERTA-CRF* was the most effective with an F1-score of 0.80.

Although our work shows promise, there are still some limitations that could be improved. One of the main challenges faced by the author is the lack of rich data provided, which limits the effectiveness of our work. We acknowledge that the accuracy of our model would improve with more data, which is a key factor in improving our results. Another significant limitation is the capabilities of our hardwares. As larger models require more resources to function properly, the hardware limitations have prevented our group from using bigger models that may be more effective. Despite these limitations, our work demonstrates potential and lays the groundwork for future improvements in this area.

In the future, we will improve our segmentation technique by populating the headings dictionary and try to integrate segmentation with NER models so that models can differentiate the university and company label. With more powerful training resources, we will do experiments on more NLP models to boost the model performance.

REFERENCES

[1] Zu, S., & Wang, X. (2019). Resume information extraction with a novel text block segmentation algorithm. Int J Nat Lang Comput, 8, 29-48.

[2] Kelkar, B., Shedbale, R., Khade, D., Pol, P., & Damame, A. (2020). Resume analyzer using text processing. Journal of Engineering Sciences, 11(5), 353-361.

[3] Amato, F., Castiglione, A., Cozzolino, G., & Narducci, F. (2020). A semantic-based methodology for digital forensics analysis. *Journal of Parallel and Distributed Computing*, *138*, 172-177.

[4] Khalid, U., Beg, M., & Arshad, M. U. (2021). RUBERT: A Bilingual Roman Urdu BERT Using Cross Lingual Transfer Learning. ResearchGate.

https://www.researchgate.net/publication/349546860_RUBERT_A_Bilingual_Roman_Urdu_BERT_Using_Cross_Lingual_Transfer_Learning

[5] A BERT-based Hierarchical Model for Vietnamese Aspect Based Sentiment Analysis. (2020, November 12). IEEE Conference Publication | IEEE Xplore.

https://ieeexplore.ieee.org/abstract/document/9287650

[6] Roy, P. K., Chowdhary, S. S., & Bhatia, R. (2020). A machine learning approach for automation of resume recommendation system. Procedia Computer Science, 167, 2318-2327.

[7] Reza, M., & Zaman, M. (2017). Analyzing CV/resume using natural language processing and machine learning (Doctoral dissertation, BRAC University).

[8] Vukadin, D., Kurdija, A. S., Delač, G., & Šilić, M. (2021). Information extraction from free-form CV documents in multiple languages. IEEE Access, 9, 84559-84575.

[9] Bhoir, N., Jakate, M., Lavangare, S., Das, A., & Kolhe, S. (2023). Resume Parser using hybrid approach to enhance the efficiency of Automated Recruitment Processes.

[10] Sinha, A. K., Amir Khusru Akhtar, M., & Kumar, A. (2021). Resume screening using natural language processing and machine learning: A systematic review. Machine Learning and Information Processing: Proceedings of ICMLIP 2020, 207-214.

[11] Bhor, S., Gupta, V., Nair, V., Shinde, H., & Kulkarni, M. S. (2021). Resume parser using natural language processing techniques. Int. J. Res. Eng. Sci, 9(6).

[12] Pai, M. Y., Chen, M. Y., Chu, H. C., & Chen, Y. M. (2013). Development of a semantic-based content mapping mechanism for information retrieval. Expert Systems with applications, 40(7), 2447-2461.

[13] Ou, X., & Li, H. (2020). YNU@ Dravidian-CodeMix-FIRE2020: XLM-RoBERTa for Multi-language Sentiment Analysis. In FIRE (Working Notes) (pp. 560-565).

[14] Shi, P., & Lin, J. (2019). Simple bert models for relation extraction and semantic role labeling. arXiv preprint arXiv:1904.05255.

[15] Yu, H., Cao, Y., Cheng, G., Xie, P., Yang, Y., & Yu, P. (2020, June). Relation extraction with BERT-based pre-trained model. In 2020 international wireless communications and mobile computing (IWCMC) (pp. 1382-1387). IEEE.

[16] Weston, L., Tshitoyan, V., Dagdelen, J., Kononova, O., Trewartha, A., Persson, K. A., ... & Jain, A. (2019). Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. Journal of chemical information and modeling, 59(9), 3692-3702.

[17] Röttger, P., Vidgen, B., Hovy, D., & Pierrehumbert, J. B. (2021). Two contrasting data annotation paradigms for subjective nlp tasks. arXiv preprint arXiv:2112.07475.

[18] Van Vinh Nguyen€, Van Long Pham*, Ngoc Sang Vu. Study of Information Extraction in Resume. https://www.semanticscholar.org/paper/Study-of-Information-Extraction-in-Resume-Nguyen-Pham/8a924b8959203689a7b3dbd60945f613708ce036#citing-papers

[19] S.Sarawagi, "Information Extraction", Foundations and Trends in Databases,vol.

1, 2008, pp 261-377.

[20] Ayishathahira C H,Sreejith C,Raseek C 2018, International CET Conference on Control, Communication, and Computing (IC4). Combination of Neural Network and Conditional Random Fields for Efficient Resume Parsing