Procesamiento del Lenguaje Natural, Revista nº 65, septiembre de 2020, pp. 53-58 recibido 19-03-2020 revisado 09-06-2020 aceptado 09-06-2020

Limitations of Neural Networks-based NER for Resume Data Extraction

Limitaciones en Reconocimiento de Entidades Nombradas (REN) basado en Redes Neuronales para la extracción de datos de Curriculum Vitae

Juan F. Pinzon, Stanislav Krainikovsky and Roman Samarev
dotin Inc. California, USA
info@dotin.us

Abstract: We provided research about the abilities of neural network-based NER models, quality, and their limitations in resolving entities of different types of complexity (emails, names, skills, etc.). It has been shown that the quality depends on the entity type and complexity, and estimate "ceilings", which model quality can achieve in the case of proper realization and well-labelled dataset.
Keywords: NER, Neural-Networks, Curriculum Vitae, Resumes, NLP, Data Extraction

Resumen: Proporcionamos investigación sobre las capacidades de los modelos REN basados en redes neuronales, la calidad y sus limitaciones para resolver entidades de diferentes tipos de dificultad (correos electrónicos, nombres, habilidades, etc.). Se ha demostrado que la calidad depende del tipo y la complejidad de la entidad, y estima los límites, que la calidad del modelo puede lograr en el caso de una realización adecuada y un corpus bien etiquetado.
Palabras clave: REN, Redes Neuronales, Curriculum Vitae, PLN, Extracción de Datos

1 Introduction

The task of parsing information from Resumes is of utmost importance and relevance in the NLP and Computer Linguistics communities. The Human Resources industry and recruiting companies will greatly benefit from its successful outcomes, not only reducing time and costs in the processing of CVs and their information but also making better quality matches between job postings and candidates. The impact and positive outcomes that these solutions can provide has been extensively discussed on research papers, such as *"The Impact of Semantic Web Technologies on Job Recruitment Processes"* (Bizer et al., 2005).

Resume Parsing solutions usually rely on complex rules statistical algorithms to correctly capture desired information from resumes. There are many variations of writing styles, words, syntax and to make things worse, the ever-increasing globalized world of nowadays means it is also important to take into account for cultural differences that affect the style and word choices among others. We intend to analyze if the Machine Learning and AI recent improvements can obtain better results for this task over rule-based approaches which use a predefined set of rules to extract the content.

Our main goal is to produce a machine learning model that will be able to extract main entities, such as name, email, education/university, companies worked at, skills, etc., from semi-structured and unstructured resumes while obtaining good quality and performance. We believe that the recent advancements in NLP can help achieve this task with smaller, but properly annotated, datasets and at the same time relying as little as possible on pre- or post-processing contrasting what was presented in the models of the "Resume Parser: Semi-structured Chinese Document Analysis" (Zhang et al., 2009) and "Study of Information Extraction in Resume" (Nguyen, Pham, and Vu, 2018) studies.

2 Methods

For this task, we used the named entity recognition (NER) approach, using the FlairNLP

ISSN 1135-5948. DOI 10.26342/2020-65-6
© 2020 Sociedad Española para el Procesamiento del Lenguaje Natural

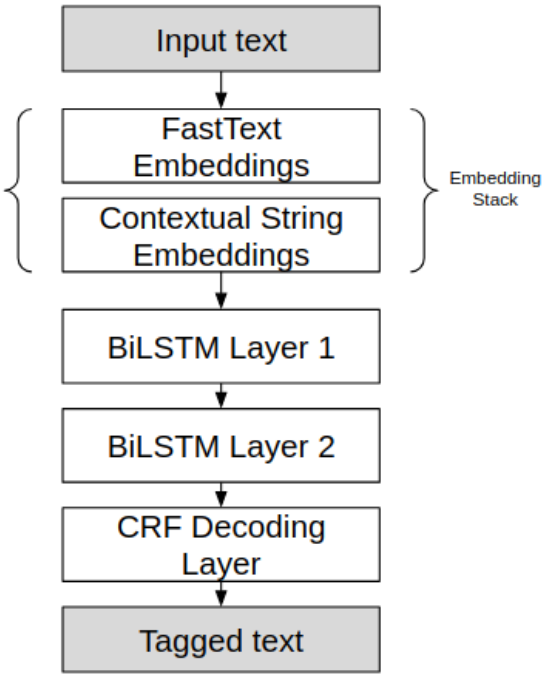

Figure 1: Neural Network NER Model Architecture

framework (Akbik et al., 2019). This framework allows us to easily implement the current state-of-the-art NER model that uses the LSTM variant of bidirectional recurrent neural networks (BiLSTMs) and a conditional random field (CRF) decoding layer, architecture depicted in Figure 1. The biggest advantage of using this framework was being able to leverage the power of Contextual String Embeddings - these embeddings can capture the hidden syntactic-semantic information, exceeding far beyond the standard word embeddings. Their distinct properties are: "(a) they are trained without any explicit notion of words and thus fundamentally model words as sequences of characters, and (b) they are contextualized by their surrounding text, meaning that the same word will have different embeddings depending on its contextual use" (Akbik, Blythe, and Vollgraf, 2018). Additionally, the stacking embeddings technique was used, which consists of combining different types of embedding models by concatenating each embedding vector to generate the final word vectors. It is proven to be beneficial to add classic word embeddings to enhance latent word-level semantics; for our task, we stacked FastText embeddings (Mikolov et al., 2018), pre-trained over web crawls, with the Flair contextual string embeddings, which are pre-trained with 1 billion word corpus.

2.1 Data

For the training of the before mentioned NER model it was required to obtain considerable amounts of annotated resumes. Two main sources of datasets with resumes were found, first one contained 220 annotated resumes from Indeed (employment website and search engine). These CVs are in a semi-structured form and have annotations for extracting 10 entities categories (Name, College Name, Degree, Graduation Year, Years of Experience, Companies worked at, Designation, Skills, Location Email Address), they were annotated collaboratively, by DataTurks community with the DataTurks online annotation tool (Trilldata-Technologies, 2018), the dataset is available in their repository (Narayanan and DataTurks, 2018). The second source was obtained from a Kaggle dataset (Palan, 2019). This dataset contains 1,200 non-annotated and unstructured resumes in a CSV file.

After analysing both sources of resumes it was discovered that the 220 annotated CVs from DataTurks were very poorly annotated, generating excessive noise on our initial NER training. For that reason, the annotations were removed and both sources were merged and re-annotated by us, using the Doccano open-source text annotation tool (Nakayama et al., 2018), initially using the same 10 entities used by DataTurks (Trilldata-Technologies, 2018).

Posterior to training several NER models, it was evident that the entity *Skills* was too broad and general and caused difficulties with the training of the NER models. The *Skills* entity was separated into *Job-Specific Skills*, *Soft Skills* and *Tech Tools*, the models were trained and compared against the 10 entities annotations, using the same resumes, improving the results.

A total of 507 resumes were annotated for each of these annotations types (10 and 12 entities) and a set-aside test set of 50 CVs (also with 10 and 12 entities annotations) were used for testing the best model. It is important to note that due to computational limitations, the resumes that had a count of tokens bigger than 2,000, were discarded before training. The corpus used to obtain the

	TRAIN	DEV	TEST
# of Documents	456	51	50
# of Tokens per Tag:			
No-tag	254,522	28,923	26,189
Name	1,479	124	170
Email_Address	2,010	227	222
Designation	6,866	670	613
Job_Specific_Skills	10,434	1,232	1,108
Tech_Tools	7,393	992	712
Companies_worked_at	5,412	524	494
Location	5,872	698	705
Years_of_Experience	6,203	696	641
College_Name	3,440	345	311
Degree	4,400	410	437
Graduation_Year	1,007	93	105
Soft_Skills	2,216	268	308

Table 1: Annotated Corpus description

highest score had the following composition of tokens and tags, see Table 1.

2.2 Cross-validated Data Quantity Analysis

A 5 fold cross-validation analysis was performed with different resumes amount, to understand the behaviour and performance of the models, as a whole (micro average) and per entity.

The second purpose was to evaluate if obtaining new annotated data will produce big enough improvements to the results. The amounts of CVs were **50**, **100**, **200**, **300**, **400** and **500**, they were taken from the Train and Development sets of annotated CVs presented in Table 1.

2.3 Best Model

The best model performance was achieved using the corpus mentioned in in Table 1 (using 12 annotated entities), the following hyperparameters and embeddings were provided to the Flair sequence tagger (NER) model:

- Embedding stack:
  - FastText word embeddings (Mikolov et al., 2018).
  - Flair Contextual String Embeddings ("news-forward") & ("news-backward"), (Akbik, Blythe, and Vollgraf, 2018).
- Initial learning rate: 0.5
- Dropout: p=0.12
- RNN layers: 2 BiLSTM layers
- Hidden size: 128

	10 Annotated Ent. F1(%)	12 Annotated Ent. F1(%)
Micro Avg.	72.1%	**78%**
Graduation Year	**92.2%**	87.2%
Name	**89.1%**	88.9%
Email	97.3%	**98.4%**
Location	87.9%	**89%**
Degree	81.5%	**85.5%**
Years Exp.	88%	**90.9%**
Designation	77.4%	**81%**
College Name	85.3%	**87.2%**
Company	67.6%	**76.1%**
Skills	56.1%	-
Job Specific Skills	-	**58.4%**
Soft Skills	-	**63.8%**
Tech Tools	-	**73.8%**

Table 2: 10 & 12 Annotated entities comparison

- Anneal factor: 0.5
- Patience: 3
- Use CRF: True

2.4 Evaluation Metrics

For the evaluation of the NER models, the metrics presented in the *"CoNLL-2003 shared task: language-independent Named Entity Recognition"* (Tjong Kim Sang and De Meulder, 2003) were used. These are exact match precision, recall and F1 score with with true-, false-positives (TP, FP) and false-negatives (FN):

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = \frac{2PR}{P + R} \tag{3}$$

Mainly the F1 (3) metrics will be presented since this measure represents a harmonic mean between Precision and Recall.

3 Results

3.1 10 & 12 Entities Comparison

The results presented in Table 2 demonstrate the success of the experiment of dividing the *Skills* entity into 3 separate entities: *Job-Specific Skills, Soft Skills* & *Tech Tools*, and therefore, helped the model in overall. Mainly, due to the difficulty and implicit nature of the *Job-Specific Skills* is localized and contained allowing for the "easier" entities like *Soft Skills* and *Tech Tools*, to be trained more precisely, thus, improving the model as a whole.

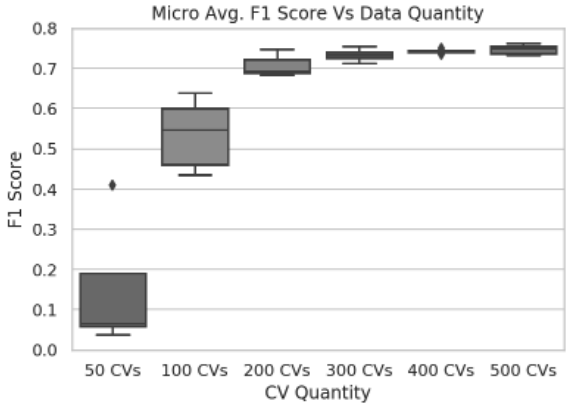

Figure 2: Micro average F1 vs. data quantity

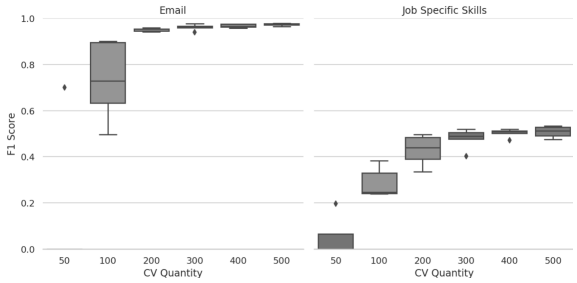

Figure 3: F1 Score vs. CV quantity for Email, Job-Specific Skills

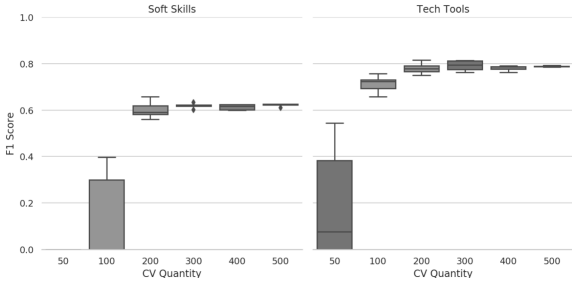

Figure 4: F1 Score vs. CV quantity for Soft Skills & Tech Tools entities

	F1 (%)	Recall (%)	Precision (%)
Micro Avg.	78.71 ±0.48	77.69	79.75
College Name	85.72 ±1.51	87.58	83.93
Company	74.21 ±2.38	78.8	70.13
Degree	85.65 ±1.53	81.86	89.8
Designation	84.08 ±1.74	91.21	77.99
Email	98.5 ±1.64	97.04	100
Graduation Year	92.73 ±2.98	92.73	92.73
Location	89.76 ±0.84	91.05	88.51
Name	89.43 ±2.28	90.16	88.71
Years Exp.	90.88 ±1.17	93.13	88.74
Job Spec. Skills	61.16 ±4.73	58.62	63.93
Soft Skills	65.27 ±2.03	60.56	70.78
Tech Tools	74.34 ±3.30	79.48	69.83

Table 3: Best model results, F1 Score, Precision and Recall

3.2 Cross-validated Data Quantity Results

Figures 2, 3 and 4 present a clear trend when more data is added for training, both for Micro average (Figure 2) and for individual entities (Figures 3 and 4). As data quantity increases the variability of the obtained F1 scores decreases. The performance of the model rapidly increases until it reaches a "ceiling", plateauing after 300 CVs. Furthermore, Figure 2 illustrates the inverse correlation between the variability of the models and training data quantity.

3.3 Best Model Results

The best model scores obtained are presented in Table 3.

4 Discussion

Figures 2, 3 and 4 demonstrate that the model reached a plateau in the scores, regarding the data quantity. Even if the labelled data is doubled, the increase in scores is going to be very limited. It might help reduce statistical variability across experiments but without much gain in the F1 score. Given this, we consider it is not worth investing the time and resources trying to increase the amount of well-labelled data.

It is important to note that the entity "nature" or complexity is not only present for the *Job-Specific Skills* entity only, Figure 3 and Table 3 illustrate how varied the scores among entities are, having different "ceilings" amongst them, there is a 37.3% difference between the highest and the lowest entities F1 scores (in best model scores). 25% (or 3 entities) present a standard deviation greater than 2.5%, *Job-Specific Skills* having the highest value (4.73%), 58.3% (or 7 entities) with a standard deviation between 2.5% and 1.5%. 25% or 3 entities present a standard deviation lower than 1.5%.

The uneven results obtained across the different entities indicate that a Neural Network NER model alone is not enough to accomplish this task with satisfactory results for production deployment. A big amount of pre- and post-processing will be required in order to identify missed entities and resolve disambiguation among the predictions. As it has been presented on previous studies by Zhang et al. (2009) and Nguyen, Pham, and Vu (2018). It can be observed that entities that are easily identifiable and are pre-

sented with the same patterns across different CV's, such as *Email Address*, *Graduation Year*, *Years of Experience* and *Location* get higher scores. Email addresses will always have a "@" sign in the middle and will finish with ".com" or ".edu" or similar. Graduation Year and Years of Experience will always be a set of start and finish date or a duration period (ex. 2004 - 2008, 4 Years, 9 Months). These easily identifiable and repeated patterns are understood by the Neural Network-based NER. Therefore, good results are obtained for these entities. On the other hand, entities which do not show any type of repeated patterns or are unique to a very small type of jobs and/or CV's, such as *Job Specific Skills*, *Soft Skills* and *Company* exhibit how this type of model fails to understand and generalize due to high amount of variability that these entities pose.

Table 3 presents how the three Skills entities present a high variation in their scores, this is also due to each of these Skills "Nature", *Job Spec Skills* being clearly the most difficult for this type of model. This is caused by the big amount of Jobs found in a diverse corpus of CV's, each different Job will have their own set of skills, making it extremely difficult for the model to understand. A solution to this would be to train different models for specific domains, using specific domain training corpus. Contrarily, results show how the entity *Tech Tools* presents a very different "Nature". Tech Tools tend to be repeated more often across CV's, even if the job category or position of the Resumes are very different. Common Tech Tools like Microsoft Word, Excel, PowerPoint, Microsoft Windows... will be found more frequently, helping the model obtain a higher percentage of total relevant results correctly classified (Recall) compared to the other skill entities.

33% of the entities obtain F1 scores below 75%, these particular entities are critical for accurate data extraction from resumes, they are *Company*, *Job-Specific Skills*, *Soft skills* and *Tech tools*. Without obtaining at least 90% F1 score for these entities it is our opinion that it can not be considered as successful results for a standalone solution. We consider 90% as a success criterion based on scores obtained by the best current commercial CV parser, Rapidparser (RapidParser, 2020). This CV parser among other commercial and open-source parsers were compared by Neumer (2018) in his Master Thesis. In this work, Rapidparser obtained F1 scores above 94% for start-date, end-date, work description and skills entities (Neumer, 2018).

NLP models have advanced greatly in recent years but still fall short when trying to implement them by themselves, especially for this task, given that the goal is to build a system that can process successfully **unstructured** and **semi-structured** CVs.

5 Conclusion

Our results demonstrate that the advancements in the NLP field, such as the "Contextual String Embeddings" (Akbik, Blythe, and Vollgraf, 2018) and state-of-the-art NER architectures have failed to obtain satisfactory outcomes to be considered useful, for HR and recruiting industry applications, on their own. The results obtained contain high variance among the different entities intended to be extracted. Obtaining good quality when entities have low complexities and are explicitly presented, such as *Email*, *Name* and *Location*, but for the more critical and complex entities such as *Job-Specific Skills* and *Soft skills* this type of model alone can not cope with the ambiguity and implicitness of this information in **unstructured** resumes.

In order to be able to meet our sponsor needs for this task, this model will be improved and enhanced with different approaches. More classical text segmentation approaches, as the ones discussed in the *"Applying named entity recognition and coreference resolution for segmenting English texts"* article (Fragkou, 2017) combined with other rule-based techniques will be tested, to resolve disambiguation of the more complex entities while at the same time trying to catch the missed ones by the model.

As a positive outcome from this project, a diverse data-set has been produced. It contains 550 CVs, 25% **semi-structured** and 75% **unstructured**, ranging from a wide variety of industries paired with well-performed annotations. This provides a unique and good quality corpus for solving this task. The corpus can be found in the following GitHub repository: https://github.com/dotin-inc/resume-dataset-NER-annotations.

References

Akbik, A., T. Bergmann, D. Blythe, K. Rasul, S. Schweter, and R. Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Akbik, A., D. Blythe, and R. Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Bizer, C., R. Heese, M. Mochol, R. Oldakowski, R. Tolksdorf, and R. Eckstein. 2005. The impact of semantic web technologies on job recruitment processes. In O. K. Ferstl, E. J. Sinz, S. Eckert, and T. Isselhorst, editors, *Wirtschaftsinformatik 2005*, pages 1367–1381, Heidelberg. Physica-Verlag HD.

Fragkou, P. 2017. Applying named entity recognition and co-reference resolution for segmenting english texts. *Progress in Artificial Intelligence*, 6, 05.

Mikolov, T., E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Nakayama, H., T. Kubo, J. Kamura, Y. Taniguchi, and X. Liang. 2018. doccano: Text annotation tool for human. Software available from https://github.com/doccano/doccano.

Narayanan, A. and DataTurks. 2018. traindata.json, Jun. https://github.com/DataTurks/Entity-Recognition-In-Resumes-SpaCy/blob/master/traindata.json.

Neumer, T. 2018. Efficient natural language processing for automated recruiting on the example of a software engineering talent-pool. Master's thesis, TECHNISCHE UNIVERSITAT MUNCHEN.

Nguyen, V. V., V. L. Pham, and N. S. Vu. 2018. Study of information extraction in resume. Technical report, VNU-UET Repository.

Palan, M. 2019. resume_dataset.csv. Kaggle, Mar. https://www.kaggle.com/maitrip/resumes.

RapidParser. 2020. Cv parsing with rapidparser: Lightning-fast! www.rapidparser.com.

Tjong Kim Sang, E. F. and F. De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Trilldata-Technologies. 2018. Dataturks - best online annotation tool to build pos, ner, nlp datasets. https://github.com/DataTurks/Entity-Recognition-In-Resumes-SpaCy.

Zhang, C., M. Wu, C.-G. Li, B. Xiao, and Z. Lin. 2009. Resume parser: Semi-structured chinese document analysis. pages 12–16, 01.