ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

# Resume: A Robust Framework for Professional Profile Learning & Evaluation

Clara Gainon de Forsan de Gabriac  Amina Djelloul
Constance Scherer  Vincent Guigue
Patrick Gallinari *

Sorbonne Université, CNRS, LIP6, F-75005 Paris, France

**Abstract**. Professional Profile Extraction is a crucial challenge for any HR department. In this paper, we propose an approach to learn and evaluate professional embeddings. We first highlight the technical issues associated with this specific data; then, we propose an architecture that compares different language models to encode the textual information; finally, we learn user profiles and propose three original evaluation tasks to illustrate the strengths and weaknesses of our approach.

## 1 Introduction

Learning to match candidates with job offers is a major challenge for any institution's human-resources department. The fast development of online job-boards (*Monster*, *JobTeaser*...) and professional social networks such as LinkedIn, makes this task increasingly crucial [1]. As such, improving profile modeling on both users and jobs may allow developing new tools to suggest relevant skills and to connect unformatted job titles and descriptions to standardized ontologies like ESCO[1].

In this work, we propose a method to learn and evaluate professional profiles using the information contained in a user's LinkedIn profile. Unlike traditional Expertise Matching methods that mainly rely on categorical data, we aim at building meaningful professional representations using only user-generated texts (their job titles and descriptions) during training in a self-supervised setting. We want our profiles to encode a sufficient amount of information to predict the future of users' careers. We also try to determine the skills and the industrial field associated with a profile.

Several technical issues make this problem difficult: job titles (and descriptions) are free texts: not only are they noisy and subject to typos, but they also prevent us from adopting a classification framework; there are as many jobs as there are users. Moreover, aggregating different jobs to build a single user profile is harder than it looks as users often mention several times the same job before obtaining a promotion. Finally, we chose a challenging framework where the final tasks –skills, industry and job predictors– are used for evaluation purposes only, without helping to refine profiles. We want to demonstrate our ability to encode relevant information in a very compact embedding from a noisy data source.

*This work is partially supported by FUI BInD.

[1] `https://ec.europa.eu/social/main.jsp?catId=1326`

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

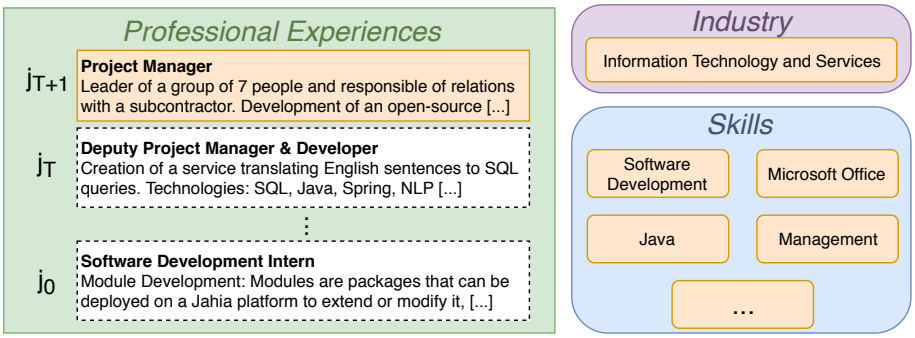


Fig. 1: User's profile schema. *A user is composed of their professional experiences, their industry and their skills. The last job, $j_{T+1}$, the industry and the skills (in orange) are the labels we use to evaluate our users' representations. Note that the skills and industry are categorical values, whereas $j_t$ (which the concatenation of both the job title and the job description) is free text.*

While text representation has long been performed at the document level in a bag-of-words setting [2], Word2vec [3] enables us to predict words in a local context opening the way for meaningful word embeddings and text generation applications. A second generation of language models introduces a solution to take into account out-of-vocabulary words through subword information encoding [4] and even more recent proposals focus on contextual embeddings and generative settings to improve sentence understanding [5]. In the field of professional profile extraction, [1] exploit a supervised framework to identify fake profiles on LinkedIn; our approach is closer to Text Summarization [6, 7] in the sense that we do not rely on supervision to learn profiles. On top of that, we also aim at generating texts to predict the next job of a particular user.

Our framework *Resume* relies on state-of-the-art robust language models to encode textual information [4, 5]. Then we aggregate job embeddings to build a user representation. On top of this framework, our main contribution resides in the evaluation approach; we were provided with more than 500k LinkedIn pages (Fig. 1) which enables us to measure quantitatively our ability to predict skills and industrial field for our set of users. We also provide an original RNN based generative approach for the next job prediction task as well as an evaluation procedure relying on a summarizing metric [8].

In this article, we introduce our models, detail the experimental settings in which we built our representations and, finally, we show that the noise level in the raw data leads to a surprising ranking of our approaches.

## 2 Models

Each user's raw data consists in a set $\mathcal{J}_u$ of chronologically ordered free texts: $\mathcal{J}_u = \{j_0, \ldots, j_T, j_{T+1}\}$. The user is also associated to a set of skills described as a binary vector in the skill domain: $\mathbf{s}_u \in \{0,1\}^S$. The industrial field is denoted $b_u \in \{1, \ldots, B\}$. Our models are composed of a job encoder that deals

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

with raw texts and a job aggregator that outputs a user profile: $\mathbf{z}_{j_t} = enc(j_t)$, $\mathbf{z}_u = agg(\mathbf{z}_{j_0}, \ldots, \mathbf{z}_{j_T})$. Then we train three independent components tackling the next job prediction task $\widehat{j}_{T+1} = dec(\mathbf{z}_u)$, the skill prediction $\widehat{\mathbf{s}}_u = f_s(\mathbf{z}_u)$ and the industrial field categorization $\widehat{b}_u = f_b(\mathbf{z}_u)$.

### 2.1 Text, Job and User Representations

We aim at representing a user's professional information using only their past jobs. Such data is noisy and contains a lot of out-of-vocabulary or rare tokens (*e.g.* product names or misspelled words). We thus choose recent text-encoding models capable of leveraging subwords information: FastText and ELMo.

*Language models.* FastText is a word embedding model based on a skip-gram formulation and optimized using negative sampling. However, it is more robust than Word2vec since it relies on subword encoding. Each character $n$-gram will correspond to a $\mathbf{z}_g$ embedding and a word embedding is simply the sum of its subwords' representations $\mathbf{z}_w = \sum_g \mathbf{z}_g$. Note that the word itself is part of its set of $n$-grams. At the word level, the skipgram formulation is implemented using a Binary Logistic Loss sliding on the text of size $T$ with a context-window $C_t$ for the word $w_t$:

$$\text{BLL}(\boldsymbol{w}) = \sum_{t=1}^{T} \left[ \sum_{c \in C_t} \ell(\mathbf{z}_{w_t}^\top \mathbf{z}_{w_c}) + \sum_{\bar{c} \notin C_t} \ell(-\mathbf{z}_{w_t}^\top \mathbf{z}_{w_{\bar{c}}}) \right], \text{with: } \ell(x) = \log(1 + e^{-x})$$

As FastText is light and easy to train, we will compare pre-trained and specifically trained embeddings on our different tasks.

ELMo is a recent language model relying on contextual embeddings: words' representations depend on their contexts. In practice, $\mathbf{z}_w$ are obtained after running a bi-directional recurrent neural network over the text: $\mathbf{z}_w$ is an aggregation of the representations of previous words until $w$ in one direction and of the following words in the other direction. As opposed to FastText, ELMo relies upon millions of parameters and we will only use the pre-trained version of this language model.

*Profile representation.* In this work, a job $j_t = (w_1^{(t)}, ..., w_N^{(t)})$ is simply encoded by averaging all its words' representations: $\mathbf{z}_{j_t} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{z}_{w_n^{(t)}}$. Then, we represent users as an aggregation of their jobs: $\mathbf{z}_u = \frac{1}{T+1} \sum_{t=0}^{T} \mathbf{z}_{j_t}$.

### 2.2 Tasks, Predictors & Decoder

We evaluate the meaningfulness of our users' representations through 3 tasks: the prediction of their skill set, the prediction of their industrial field and the generation of their last job. We refer to them as the Skill Predictor, the Industry Predictor, and the Last Job Decoder.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

*Predictors.* Since we aim at evaluating our embeddings' quality, we made the predictors simple. Both the Skill Predictor $f_s$ and the Industry Predictor $f_b$ consist of two linear layers separated with tanh activation functions and followed by a sigmoid for the former and a softmax for the latter:

$$f_s(\boldsymbol{z}_u) = \text{sigmoid}(W_2^\top(\tanh(W_1^\top \boldsymbol{z}_u))), \quad f_b(\boldsymbol{z}_u) = \text{softmax}(U_2^\top(\tanh(U_1^\top \boldsymbol{z}_u))).$$

When predicting a user's skill set, we are interested in predicting all the right skills and only the right skills, placing us in a multi-label classification task. This predictor is trained to optimize a Binary-Cross-Entropy Loss function. The industry prediction is a simple mono-label classification task. The Industry Predictor is trained to optimize a Cross-Entropy Loss function.

*Decoder.* The Last Job prediction, however, is a harder task: it cannot be addressed as a classification task since the number of unique jobs is in the same order of magnitude as the number of users. Thus, we choose to generate the title and the description of a user's last job $j_{T+1}$ from his representation $\boldsymbol{z}_u$. In this context, we implemented a LSTM, followed by a linear layer (of weights $V$), that decode $\boldsymbol{z}_u$ into a sequence of words $\widehat{j}_{T+1}$. At each time step, we feed the decoder $\text{dec}(\boldsymbol{z}_u, w_n^{(T+1)})$ both the user representation $\boldsymbol{z}_u$ and the last-predicted token $w_n^{(T+1)}$ and it outputs the next token $\widehat{w}_{n+1}^{(T+1)} = V^\top \text{LSTM}([\boldsymbol{z}_u, w_n^{(T+1)}])$. It is trained to optimize a Cross-Entropy Loss function between the predicted word $\widehat{w}_n^{(T+1)}$ and the label $w_n^{(T+1)}$, for every word in the actual sequence.

## 3 Experiments and Results

In this section, we present our results and analyze them. After giving all implementation details to ensure reproducibility, we propose to challenge our models with relevant baselines. We chose a strong and simple approach based on the most common classes: we always predict the most common industrial field (*"Marketing and Advertising"*) and the $M$ most frequent skills, $M$ being the averaged number of skills per profile ($M = 10$). Similarly, we propose the $N$ most common words as a baseline for the last job prediction ($N = 41$).[2]

### 3.1 Dataset and Training Details

Our dataset consists of over 500,000 LinkedIn users' pages, and almost 5,000,000 professional experiences once we eliminate profiles with less than 3 jobs. There are more than 95,000 different skills cited in the dataset but we only retained those appearing in at least 3000 profiles, leaving us with 523 skill classes. Similarly, we kept 147 industrial fields, also referred to as industries.

Our 3 user representations rely respectively on a FastText model trained on our data $FT_{CV}$, a pre-trained FastText $FT_{pt}$[3] and a pre-trained ELMo.[4]

[2]Resulting sentences are unintelligible but it should be very strong to retrieve unigrams (in particular for the stopwords).

[3]`https://fasttext.cc/docs/en/crawl-vectors.html`

[4]`https://github.com/allenai/allennlp/blob/master/tutorials/how_to/elmo.md`

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

Both the Skill Predictor and the Industry Predictor were trained on a hundred epochs with an Adam optimizer (with the recommended hyper-parameters) and a batch size of 160. Both models have a hidden layer size equal to the input layer size. The Last Job Decoder was trained with a batch size of 100 and a vocabulary restricted to the 40,000 most frequent words of our dataset.

For this latter task, we evaluate our results using the BLEU metric which measures the quality of a textual prediction with respect to a ground truth [8].

### 3.2 Results and Analysis

The results of our classification tasks on skills and industry prediction and those of text generation are reported in Table 1 and Table 2 respectively. Those results show that the $FT_{CV}$ outperforms the other models in both skills and industry prediction, as well as in text generation. This ranking among our approaches is surprising, it points out that the CV style does not follow a classical language model: a (very) robust and dedicated model is required to tackle misspellings, abbreviations, ellipses, acronyms that characterize the fast writing style observed on CV. The same kind of conclusions has been drawn in [9].

Such results, while initially counter-intuitive, can be empirically understood when taking a closer look at the predictions. The last job generation highlights the difficulty for our models to generate long job descriptions as well as very specific sentences. For instance, the encoding-decoding process gives:

GT: *Title: E-commerce Consultant, Desc: My mission consists in reaching the goals set up by the clients regarding their profitability and/or notoriety issues [...]*

$FT_{pt}$ *Title: Marketing Manager, Desc: Management of the client relationship, Social networks management, Social networks management [...]*

$FT_{CV}$ *Title: Marketing Manager, Desc: Managing the communication strategy and the communication strategy for clients[...]*

ELMo *Title: Secteur Manager, Desc: Management of the client relationship, [UNK], stock management, stock management [...]*

Such predictions highlight the complexity and diversity of our data. While not human-like, those predictions can add a lot of value to a CV database as they capture the essence of a career. The analysis of both skills and industry prediction for all three models indicate that a consequent part of the wrong predictions make sense to a human reader. For instance, a profile containing the skills *Office Pack*, *Photoshop* and *Marketing* is predicted to have the *Microsoft Word*, *Adobe Photoshop* and *Digital Marketing* skills. Similarly, a Developer working in the Pharmaceutical Industry can be either predicted in the "IT" or the "Pharmaceutical" industry. Those observations lead us to believe that the representation of our users is rather satisfactory. Most prediction errors are understandable and could be tackled by a more thorough data pre-processing.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

| Model | F1 score - Skills Prediction | Accuracy - Industry Prediction |
|---|---|---|
| Most Common | 24.0% | 6.3% |
| $FT_{pt}$ (pre-trained FastText) | 40.9% | 35.6% |
| $FT_{CV}$ (CV-oriented FastText) | **42.4%** | **38.4%** |
| ELMo | 39.0% | 30.7% |

Table 1: Experimental results on the classification tasks.

| Model | Bleu score (Last Job) | | | | |
|---|---|---|---|---|---|
| | Bleu | Bleu-1 | Bleu-2 | Bleu-3 | Bleu-4 |
| Most Common | 0.00 | **33.3** | 0.3 | 0.0 | 0.0 |
| $FT_{pt}$ (pre-trained FastText) | 1.91 | 20.6 | 3.5 | 0.8 | 0.2 |
| $FT_{CV}$ (CV-oriented FastText) | **2.15** | 22.2 | **3.8** | **0.9** | **0.3** |
| ELMo | 1.74 | 22.5 | **3.8** | 0.6 | 0.2 |

Table 2: Experimental results on job prediction (title & description).

## 4 Conclusion and Discussion

We propose a novel approach to professional profile learning, Resume, that is self-supervised and relying on free text, along with three evaluation tasks. We compare the impact of the use of a language model to the use of a word embedding model on our evaluation tasks. Our experiments show that using a word embedding model to represent users is not only sufficient to modelize their professional information but also outperforms heavier language models architecture on all of our tasks. In future work, we want to address the representation of even more complex entities, such as companies and especially the prediction of even more elaborate signals, like fund-raising or company expansion. Another application could be a job training model, helping users reach their career objectives by indicating the skills they should acquire.

## References

[1] Shalinda Adikari and Kaushik Dutta. Identifying fake profiles in linkedin. In *PACIS*, 2014.

[2] David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003.

[3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, 1310.4546, 2013.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *CoRR*, 1607.04606, 2016.

[5] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *CoRR*, 1802.05365, 2018.

[6] Inderjeet Mani. *Advances in Automatic Text Summarization*. MIT Press, 1999.

[7] R. Paulus, C. Xiong, and R. Socher. A deep reinforced model for abstractive summarization. *CoRR*, 1705.04304, 2017.

[8] T. Ward K. Papineni S. Roukos and W. Zhu. Bleu: A method for automatic evaluation of machine translation. In *ACL*, 2002.

[9] Pallavi Jain, Robert Ross, and Bianca Schoen-Phelan. Estimating distributed representation performance in disaster-related social media classification. In *ASONAM*, 2019.