JobBERT:
Understanding Job Titles through Skills

Jens-Joris Decorte[1,2], Jeroen Van Hautte[2],
Thomas Demeester[1], and Chris Develder[1]

[1] Ghent University – imec, 9052 Gent, Belgium
{jensjoris.decorte, thomas.demeester, chris.develder}@ugent.be
https://ugentt2k.github.io/
[2] TechWolf, 9000 Gent, Belgium
{jensjoris, jeroen}@techwolf.ai
https://techwolf.ai/

Abstract. Job titles form a cornerstone of today's human resources (HR) processes. Within online recruitment, they allow candidates to understand the contents of a vacancy at a glance, while internal HR departments use them to organize and structure many of their processes. As job titles are a compact, convenient, and readily available data source, modeling them with high accuracy can greatly benefit many HR tech applications. In this paper, we propose a neural representation model for job titles, by augmenting a pre-trained language model with co-occurrence information from skill labels extracted from vacancies. Our JobBERT method leads to considerable improvements compared to using generic sentence encoders, for the task of job title normalization, for which we release a new evaluation benchmark.

Keywords: Job Title Normalization · Semantic Text Similarity · Information Extraction.

1 Motivation and Related Work

Job titles form a crucial piece of textual information within the job market. For job seekers, they are a primary way of searching for relevant job postings. In HR departments of large organizations, they serve as an accessible data source for gathering insights or taking actions concerning their in-house talent. Although job titles generally adhere to a certain structure [12], they are often freely entered, unstructured texts. Nevertheless, standardized lists of job titles have been established, e.g., in the European Skills, Competences, Qualifications and Occupations (ESCO) taxonomy [5]. The task of relating free-form job titles to their most relevant standardized job title is called job title normalization and is a crucial step towards valuable information extraction (IE) from HR data.

Job title normalization has traditionally been approached as a supervised learning problem. A first effort by LinkedIn defined a "couple dozen" standard job categories and performed classification based on common key phrases [2].
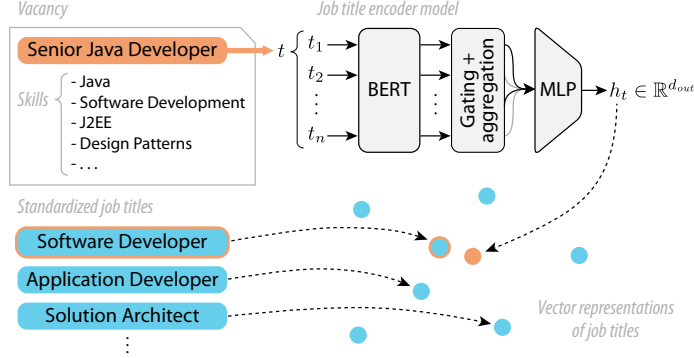
Fig. 1. Our JobBERT model uses a BERT-based encoder to obtain vector representations of job titles. Matching a job title to standardized job titles is based on proximity in the representation's vector space. Representations are trained based on vacancy titles and their associated skills: titles with similar skills are encoded in similar vectors.

A more elaborate taxonomy of over 4k job titles was used in Carotene, using a hierarchical SVM-kNN cascade model for classification [6]. The same task was more recently solved by DeepCarotene, using an end-to-end deep convolutional neural network [13]. This supervised approach suffers from three disadvantages:

(a) It requires an extensive set of job titles that are labeled with their normalized counterpart. The list of standardized job titles often has multiple thousands of items, making this annotation step extremely difficult without deeply familiar expert knowledge.
(b) To keep the model up to date with trends in the job market, a continuous data labeling effort is required.
(c) The resulting system is inherently tied to an a priori defined (and thus static) list of standardized titles. This limits its practical usability as it cannot be customized or tuned, for example towards specific HR department needs.

The main contribution of this paper is a novel method for job title normalization that overcomes the above-mentioned problems (a)–(c). We achieve this by approaching the task as a semantic text similarity (STS) task in which semantically similar job titles should lie close to each other in a learned vector space, as illustrated in Fig. 1. The resulting representations allow the usage of distance-based metrics for normalizing job titles with respect to *any* job title taxonomy, thus alleviating problem (c). This idea is similar to the approach taken in [10], where a Siamese LSTM model is trained on explicit pairs of similar job titles. Although effective, their method still suffers from problems (a) and (b) as it requires a large handcrafted dataset of synonymous job titles, along with hand-picked augmentation rules. To bypass the need for labeling training data, we propose a distant supervision setup in which our semantic job title encoder learns to understand the meaning of job titles through the *skills* they rely on, inferred from vacancies including descriptions thereof. This approach and the

reasoning behind it are further detailed in §2. Finally, we construct an evaluation dataset for the job title normalization task and report several performance metrics for our method as well as other commonly used sentence encoding techniques. We conclude that our JobBERT method outperforms Sentence-BERT (SBERT), which represents the state-of-the-art for general STS tasks [11]. To foster follow-up research on the task of job title normalization, we release the evaluation dataset to allow benchmarking.

2   Job Title Representation Learning

Our method for constructing a representation h_t for a job title t is built upon the premise that skills are the essential components defining the job. As such, semantically similar job titles should be similar in terms of the skills they require. We apply this idea to our model by optimizing it to predict the skills that represent a given job title. To bring this method into practice, a large corpus of titles and their corresponding set of skills is needed: we assume a vocabulary of skills $s \in V$, and a set of titles $t \in T$ each with their corresponding set of skills $S_t \subset V$. Although this might seem like a hard requirement at first, this is easy in practice due to the generally high availability of *vacancies* posted online. Vacancy titles are generally representative of the real distribution of job titles in the job market. Their description contains information about the skills that are required for the job. To avoid the costly manual tagging of skills within vacancies, we apply a simple distant supervision rule to create a noisy training set of vacancy titles with associated skills, through literal string matches in the vacancy texts with an extensive proprietary reference list of skills.

What follows is a description of our model architecture and optimization procedure. The main part of the model architecture is the job title encoder, as outlined in Fig. 1. It consists of the BERT model [3], a weighted averaging operation and a fully connected output network (MLP). The final distributed representation $h_t \in \mathbb{R}^{d_{out}}$ for job title t is calculated as shown in eq. (1). Newly introduced notations are explained below, along with the corresponding parts of the encoder network.

$$h_t \;=\; \text{MLP}\left(\sum_{i=1}^{n} \frac{\sigma\left(w_g \cdot \text{BERT}\left(t_i\right) + b_g\right)}{\sum_{j=1}^{n} \sigma\left(w_g \cdot \text{BERT}\left(t_j\right) + b_g\right)} \, \text{BERT}\left(t_i\right)\right) \tag{1}$$

Pre-trained Language Model We use BERT as the backbone of our encoder network. For efficiency reasons, we use the BERT-base model. It uses the WordPiece tokenizer to split an input text t into a sequence of sub-word tokens $t_1, t_2, \ldots, t_n$, where n depends on the length of the input [14]. This model outputs a d-dimensional vector for each input token t_i, denoted by $\text{BERT}(t_i)$.

Gating and Aggregation To obtain a fixed-length job title representation, the output vectors of BERT have to be aggregated. The authors of [11] show that

averaging the token embeddings yields the best performance in their Sentence-BERT model, so we take this as our starting point. However, we should note that not all words in a vacancy title provide information about the job content. As an example, the title *"software developer London"* should lie close in the vector space to *"software dev. in Ghent"*, but far from *"business developer London"*. In other words, the model should learn to attribute more weight to meaningful words while neglecting indicators such as locations that are irrelevant with respect to job title classification. To this end, we add a gating mechanism that acts as an appropriate inductive bias to allow the model to ignore particular words before the aggregation into a single vector representation. The gating coefficient for each token position i is calculated as the weighted sum $(w_g \cdot \text{BERT}(t_i) + b_g)$ of all components in the token representation $\text{BERT}(t_i)$, with the weight vector w_g and bias b_g, followed by a sigmoid nonlinearity. These gating coefficients are then normalized by their sum, after which they are used to aggregate all token representations into a single weighted sum.

Projection MLP The last part of the encoder network is the projection multilayer perceptron (MLP), which consists of two linear layers with a ReLU activation in between. It transforms the aggregated d-dimensional embedding into the final representation $h_t \in \mathbb{R}^{d_{out}}$.

Context Matrix and Optimization The encoder network is trained to predict the bag of skills S_t that corresponds to a given title t in the training data. To formalize this, we draw inspiration from the Skip-Gram model with negative subsampling [9]. For each positive training pair (t, s_j), with $s_j \in S_t$ a skill associated with job title t, we randomly draw K negative skills from a noise distribution. This distribution is defined by the frequency distribution of the skills in the training corpus raised to the power of $3/4$, as this has been shown to work well [9]. The model is optimized to predict the presence of the positive label s_j while predicting the absence of the K negative labels. To facilitate this optimization, the model contains a trainable context matrix $U \in \mathbb{R}^{|V| \times d_{out}}$ that consists of context vectors $u_i \in \mathbb{R}^{d_{out}}$ for each skill s_i in the skill vocabulary V.[3] The log-likelihood of each positive pair (t, s_j) (with $s_j \in S_t$) is given by:

$$\mathcal{J}(t, s_j) = \log \sigma(u_j \cdot h_t) + \sum_{k=1}^{K} \log \sigma(-\tilde{u}_k \cdot h_t) \qquad (2)$$

In the above equation, $\{\tilde{u}_k \mid k = 1 \dots K\}$ are the context vectors corresponding to the K negative sampled skills. The gradient updates for both the encoder model and the context matrix are calculated with the objective of minimizing the negative log-likelihood.

---

[3] Note that the context matrix is not shown in Fig. 1 as it is only needed during training and can be discarded at inference time.

3   Experimental Setup

Benchmark Dataset In order to evaluate the performance of our model, we create and publish a new dataset of vacancy titles that are labeled with the standardized ESCO occupations.[4] This data was gathered from a large governmental online job board, on which each job posting is tagged with the most suitable ESCO occupation label by its creator.[5] The ESCO occupations are organized within a hierarchical structure. For example, both *"aircraft engine assembler"* and *"wave soldering machine operator"* are leaf nodes in the sub-tree of the generic *"assembler"* occupation. To have a mutually exclusive label space, we reduce the hierarchy of ESCO occupations into a list of its leaf nodes, after which 2,675 out of the total of 2,942 occupations remain. We discard all non-English vacancy titles in the dataset by applying the fastText Large language identification model [7,8]. The final dataset contains a total of 30,926 unique English vacancy titles, each tagged with one of the remaining ESCO occupations. We make a stratified split of this dataset based on the ESCO label distribution into two equally sized parts, for validation and testing. We note that not all vacancy titles in this dataset are equally descriptive. Hence, from analyzing a random subsample of 2,000 vacancies in this dataset, we estimate an upper bound of 65% of the vacancy titles that can be linked to their corresponding ESCO label without ambiguity.

Training Details We describe the high-level experimental setup here and refer to Appendix A for more details. We select a large training corpus of 300 million English vacancies from TechWolf's internal Data Lake. A proprietary vocabulary of over 35k clean skill names is used to generate the skill tags for self-supervision. Vacancies are tagged with skills based on literal occurrence in the text.[6] This is a rather simple skill extraction method that can be improved upon in many ways. However, as a distant supervision approach applied on a large dataset, it is qualitative enough to provide strong results.

The BERT model is initialized with its pre-trained weights. We choose an output dimension of $d_{out} = 300$, which is much smaller than the original 768-dimensional BERT embeddings. This is beneficial for the application of job title normalization as it reduces storage requirements and speeds up the calculation of similarity metrics. We use a batch size of 64 (title, skill)-pairs during training. The number of negatives appeared to have little impact on the performance and we set it to $K = 5$. In our first experiment, the pre-trained BERT weights are kept fixed. Afterwards, we investigate the further improvements to be made by also finetuning the BERT weights. Throughout the training, the performance is measured on the validation set in terms of mean reciprocal rank (MRR) after each multiple of 5k batches. We stop training when the validation score shows no further improvements, which happens after the first 8.5% of the fist epoch.

---

[4] https://github.com/jensjorisdecorte/JobBERT-evaluation-dataset

[5] https://www.myfuturejobs.gov.my/

[6] Casing is ignored, except for skills that contain a majority of uppercase characters.

4   Results and Analysis

We compare our JobBERT model with most commonly used sentence representation models LASER [1], fastText [8] and Sentence-BERT [11]. For the Sentence-BERT model, we used its base model with averaged token embeddings ($SBERT_{AVG}$). We also included the plain BERT-base model [3] with averaged token embeddings ($BERT_{AVG}$). We calculate both MRR and Recall@k metrics on the test set by ranking the ESCO labels based on cosine similarity with the input title. In Table 1 we report both macro and micro averages because of the long tail class distribution in the test dataset. Our results confirm the findings of [11] that averaging BERT token embeddings does not yield good representations, as they are surpassed by both fastText and LASER representations. However, our JobBERT approach outperforms Sentence-BERT, even when keeping the pretrained BERT weights fixed, while finetuning (FT) BERT further considerably increases the test scores.

We finally note that our JobBERT model can be inspected by analyzing the gating coefficients for each token in a given job title. Such analysis in Appendix B demonstrates that the model successfully neglects words that are irrelevant for job title classification (e.g., locations).

Method	MRR		Recall@1		Recall@5		Recall@10	
	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro
LASER [1]	0.2194	0.2250	0.1600	0.1723	0.2596	0.2537	0.3046	0.2960
fastText [8]	0.2533	0.2313	0.1710	0.1602	0.3217	0.2823	0.3918	0.3443
$BERT_{AVG}$	0.2000	0.2056	0.1355	0.1486	0.2426	0.2396	0.2963	0.2843
$SBERT_{AVG}$	0.2695	0.2648	0.1863	0.1930	0.3426	0.3262	0.4160	0.3801
JobBERT[*]	0.3261	0.2690	0.2307	0.1919	0.4207	0.3352	0.4991	0.3993
$JobBERT_{FT}$[*]	**0.3641**	**0.3092**	**0.2666**	**0.2248**	**0.4632**	**0.3865**	**0.5440**	**0.4604**

[*] : Our proposed methods; FT = with finetuning the BERT weights.

Table 1. Comparison of representation-based job title classification methods.

5   Conclusion

We present JobBERT, a novel approach for the task of job title normalization, which builds upon the premise that skills are the essential components defining a job. We rely on distant supervision by learning to represent vacancy titles based on skills included in their description. This eliminates the need for manual data labeling, which is costly, especially to keep the model up to date with rapidly changing trends in the job market. The strength of the JobBERT representations is demonstrated by applying them for the task of job title normalization through mapping on a taxonomy not seen during training of the model. The task is cast as a ranking problem, and we achieve improvements of the order of +10% in mean reciprocal rank as well as Recall@10 compared to state-of-the-art encoders for semantic text similarity.

6   Acknowledgments

References

1. Artetxe, M., Schwenk, H.: Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. Trans. Assoc. Comput. Linguistics **7**, 597–610 (2019), `https://transacl.org/ojs/index.php/tacl/article/view/1742`
2. Bekkerman, R., Gavish, M.: High-precision phrase-based document classification on a modern scale. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 231–239 (2011)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, `https://www.aclweb.org/anthology/N19-1423`
4. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research **12**(61), 2121–2159 (2011), `http://jmlr.org/papers/v12/duchi11a.html`
5. ESCO: European skills, competences, qualifications and occupations. EC Directorate E (2017)
6. Javed, F., Luo, Q., McNair, M., Jacob, F., Zhao, M., Kang, T.S.: Carotene: A job title classification system for the online recruitment domain. In: 2015 IEEE First International Conference on Big Data Computing Service and Applications. pp. 286–293. IEEE (2015)
7. Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T.: Fasttext.zip: Compressing text classification models. arXiv preprint arXiv:1612.03651 (2016)
8. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 427–431. Association for Computational Linguistics, Valencia, Spain (Apr 2017), `https://www.aclweb.org/anthology/E17-2068`
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. p. 3111–3119. NIPS'13, Curran Associates Inc., Red Hook, NY, USA (2013)
10. Neculoiu, P., Versteegh, M., Rotaru, M.: Learning text similarity with siamese recurrent networks. In: Proceedings of the 1st Workshop on Representation Learning for NLP. pp. 148–157 (2016)
11. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2019), `https://arxiv.org/abs/1908.10084`

12. Van Hautte, J., Schelstraete, V., Wornoo, M.: Leveraging the inherent hierarchy of vacancy titles for automated job ontology expansion. In: Proceedings of the 6th International Workshop on Computational Terminology. pp. 37–42. European Language Resources Association, Marseille, France (May 2020), `https://www.aclweb.org/anthology/2020.computerm-1.5`
13. Wang, J., Abdelfatah, K., Korayem, M., Balaji, J.: Deepcarotene-job title classification with multi-stream convolutional neural network. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 1953–1961. IEEE (2019)
14. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., Dean, J.: Google's neural machine translation system: Bridging the gap between human and machine translation. CoRR **abs/1609.08144** (2016), `http://arxiv.org/abs/1609.08144`

Appendix A   Training details

During training, we keep all (title, skill)-pairs originating from the same vacancy together in one batch and pass each unique title only once through the encoder model. Given that an average vacancy in our training data contains around 14.7 literal skill mentions, this trick speeds up the forward pass during training by over an order of magnitude. We use a relatively large batch size of 64 which makes the training procedure robust to noise introduced by the distant supervision.

In the first experiment, we initialize BERT with its pre-trained weights and keep those fixed. The context matrix U is initialized with zero values, in accordance with the original implementation of the Skip-Gram model [9]. The token gating and the MLP both consist of linear layers. Each of those linear layers — with input dimension I and output dimension O — are initialized as follows:

1. The *weight* parameters are randomly initialized from a uniform distribution in the range of $\left(-\sqrt{\frac{1}{I}}, \sqrt{\frac{1}{I}}\right)$,
2. The *bias* parameters are randomly initialized from a uniform distribution in the range of $\left(-\sqrt{\frac{1}{O}}, \sqrt{\frac{1}{O}}\right)$.

We use a simple stochastic gradient descent (SGD) optimizer with a learning rate of 0.05 for both the gating mechanism and MLP weights. The context matrix is optimized with the Adaptive optimization scheme Adagrad [4] with the default learning rate of 0.01. This optimizer tracks which skills occur more frequently during training and reduces the learning rate for those, which stabilizes the training procedure.

In the second experiment, the BERT weights are finetuned. In order to stabilize the training procedure, we initialize the model with the weights from the first experiment. The learning rate of the SGD optimizer is reduced to 0.001 as it is now also used to update the BERT weights.

For the fixed BERT model, we see no significant performance improvement on the evaluation set after training on 15 million vacancies. Further improvements

from finetuning the BERT weights stall after training on an additional 10.5 million vacancies. As such, we train for less than one epoch in total, on only the most recent 8.5% of vacancies. All experiments were run on an NVIDIA K80 GPU.

Appendix B   Analysis of the Gating Coefficients

The token gating mechanism in our JobBERT model allows the inspection of which tokens in a job title the model perceives to be important for its meaning. Doing so reveals that the model successfully ignores tokens that do not communicate information regarding the job content. Below, some examples are illustrated. Each token in the input job title is accompanied by its gating coefficient. The coefficient values range from zero (not important) to one (important) and are reflected in the intensity of the red background color:

1. elementary$_{(.77)}$ school$_{(.84)}$ teacher$_{(.95)}$ french$_{(.35)}$ part$_{(.02)}$ -$_{(.01)}$ time$_{(.07)}$.

2. AI$_{(.97)}$ specialist$_{(.51)}$ in$_{(.07)}$ London$_{(.18)}$.

3. head$_{(.43)}$ of$_{(.26)}$ data$_{(.90)}$ science$_{(.80)}$ in$_{(.09)}$ our$_{(.13)}$ head$_{(.11)}$ quarters$_{(.09)}$.

Examples 1 and 2 illustrate how meaningless indicators such as "in London" or "part-time" are ignored by the model, while the important tokens receive high gating coefficients. Example 3 contains the "head" token twice, but attributes a four times higher gating coefficient to the first appearance which adds to the meaning of the job title.