Steffen Schnitzer, Dominik Reis, Christoph Rensing, Wael Alkhatib, Ralf Steinmetz: *Preselection of Documents for Personalized Recommendations of Job Postings based on Word Embeddings.*
In: The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3297280.3297602

Preselection of Documents for Personalized Recommendations of Job Postings based on Word Embeddings

Steffen Schnitzer
Dominik Reis
steffen.schnitzer@kom.tu-darmstadt.de
dominik.reis@gmail.com
TU Darmstadt
Multimedia Communications Lab
Darmstadt, Germany

Wael Alkhatib
Christoph Rensing
Ralf Steinmetz
wael.alkhatib@kom.tu-darmstadt.de
christoph.rensing@kom.tu-darmstadt.de
ralf.steinmetz@kom.tu-darmstadt.de
TU Darmstadt
Multimedia Communications Lab
Darmstadt, Germany

ABSTRACT

In the search for matching jobs, more and more people rely on online services such as job search engines. Job search engines provide the possibilities of searching for certain keywords and maintaining domain related filters like the location or the seniority of a job posting. A job recommendation system can support the users on such platforms by finding relevant jobs that match their profile. When it comes to job postings, the platform often has no information about whether a user actually applied for a certain job or whether the application was successful. In this paper, we propose a method to use the implicit information that users provide on the platform to recommend matching job postings in real time. We provide a solution by applying the *doc2vec* method on the job descriptions to cluster them. This allows us to preselect certain job postings and reduce the target space to implement a personalized classifier for recommendation. Both the quality of recommendations and the runtime of the according algorithms are improved. Our evaluation with domain experts shows, that at least 55% of these recommendations are relevant to the respective user.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Clustering and classification**; *Document filtering*; • **Computing methodologies** → Machine learning approaches;

KEYWORDS

Preselection, Word Embeddings, Job Recommendation

ACM Reference Format:
Steffen Schnitzer, Dominik Reis, Wael Alkhatib, Christoph Rensing, and Ralf Steinmetz. 2019. Preselection of Documents for Personalized Recommendations of Job Postings based on Word Embeddings. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3297280.3297602

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SAC '19, April 8–12, 2019, Limassol, Cyprus
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5933-7/19/04.
https://doi.org/10.1145/3297280.3297602

1 INTRODUCTION

Due to the high availability of the Internet, the job market has also spread to platforms on the World Wide Web. Various online job exchanges, job-oriented social networks, and job search engines establish themselves with the aim to offer each job seeker the job offerings that best suit their interests.

A very high number of job postings can be found online. Due to the high number of available job vacancies, the job seeker requires digital support in finding relevant job postings [8]. Accordingly, a recommendation system can be very useful for a job seeker [7].

In order to create a user profile, feedback from the user is required. In general, there is a distinction between explicit and implicit feedback. With explicit feedback, the user explicitly evaluates objects positively or negatively, as for example in a five star rating system. Implicit feedback is obtained indirectly through objects clicked on by the user, which can be implied as positive feedback.

The goal of the methods presented in this paper is to develop a personalized recommendation system for job postings, based on implicit feedback.

To reach the goal of providing a personalized recommendation of job postings using implicit feedback from job seekers on a job search platform, the content of the unstructured job descriptions is analyzed using the word embeddings based *doc2vec* method [4] for feature generation. The job postings are clustered according to these features. The clusters allow us to propose a novel approach for a personalized preselection of web documents, namely the job postings. From the resulting preselected documents, a classifier decides which jobs to recommend to the user. Our preselection approach improves the quality of recommendations as well as the runtime of the corresponding algorithms.

Two different approaches, used for the final recommendation, are described in addition to the preselection. On the one hand, an approach based on the work of Huang [3] using an averaged KNN for recommendation is used as a baseline and improved by introducing the preselection. On the other hand, an approach based on the work of Vuurens et al. [12] using a neural network learned hyperplane is presented. The results are evaluated by an offline evaluation on a representative sample of approximately 2 million job postings and more than 300,000 users in terms of Precision@10. Furthermore, we compare our approaches against two relevant baselines [2, 13], which apply a distributed KNN approach. Due to

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, not withstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

the runtime of these approaches, an evaluation on representative samples of the dataset is provided. As the evaluation on implicit user ratings on a huge corpus provides very small values for the precision, an additional evaluation by domain experts is provided, to judge the overall applicability of the presented approach.

2 RELATED WORK

Tripathi et al. [11] present an overview of different job recommendation procedures. They provide many different methods for calculating recommendations based on a wide variety of starting points. The job postings in our dataset can be viewed as unstructured texts respectively web documents [8, 9].

Melville et al. [5] report that if on average more than 99.5% of the objects are not rated by users, the quality of collaborative approaches is drastically reduced. Since this value is at 99.9981% in our dataset of job-postings, collaborative approaches are not considered.

The content-based method by Poch et al. [7] is used to find suitable jobs on the *Job Talent* platform. They process a personal textual description of the user using various NLP methods. For each user, only the job postings from an assigned cluster have to be ranked. This preselection idea serves as a basis for our own approach, as it improves both performance and runtime. Huang [3] develop a content-based recommendation system for job postings on Xing. The job postings are ranked using an averaged KNN approach [6]. Applying *doc2vec* and the averaged KNN approach, we refer to this baseline as *D2V-AVGKNN*. Zhang and Cheng [13] present an ensemble method using a content-based and a collaborative approach. The content-based procedure creates feature vectors in the same way as Huang [3] using *doc2vec*, but only on the basis of titles and tags. The recommendations are made using a distributed KNN algorithm. Applying *doc2vec* and a distributed KNN approach, we refer to this baseline as *D2V-DSTKNN*. Guo et al. [2] present different text-based procedures on the *Careerbuilder* job exchange. They create their feature vectors in two different ways: Using a bag-of-words approach, and on the basis of entities such as persons, companies, and locations. Applying the bag-of-words method and a distributed KNN approach, we refer to this baseline as *BOW-DSTKNN*. Vuurens et al. [12] rank movies and books according to user preferences. Their work is based on the article of Musto et al. [6] and uses the *doc2vec* method to train the document vectors on the basis of the Wikipedia entries of the corresponding movies. With the help of a neural network, a hyperplane is calculated for each user whose normal vector can be used to determine the ranking. This method serves as the basis for our hyperplane approach presented in this paper.

Based on these different approaches, we create our approach applying *doc2vec* on title and descriptions of job postings. On the one hand, we combine our preselection of positive documents with an averaged KNN approach (*PRS-D2V-AVGKNN*). On the other hand, we provide positive and generate negative training samples using the preselection for a hyperplane approach (*PRS-D2V-HP*).

3 METHODOLOGY

As user and document data we define the set of job posting documents $\tilde{d} \in \tilde{D}$, the set of users $\tilde{u} \in \tilde{U}$, as well as the corresponding ratings ($\tilde{L}$). Basic text processing is used on the raw documents $\tilde{D}$, resulting in the cleaned documents $\bar{D}$, which are handed to the *doc2vec* model, which projects the text-based objects into a feature space $D \subseteq \mathbb{R}^{\phi_f}$ with ϕ_f feature dimensions.

To create the user profile, the documents are grouped into ϕ_k clusters $C_1, C_2, \ldots, C_{\phi_k}$, which are used for preselecting documents for each user. The aim is to evaluate the different clusters for each user according to their preferences. This results in a subset of positive preselected documents $\tilde{S}^+ \subseteq \tilde{D}$ and a subset of negative preselected documents $\tilde{S}^- \subseteq \tilde{D}$. The classification calculates the user profiles based on the preselected documents and the rated job postings of the users, which projects the users into the feature space $U \subseteq \mathbb{R}^{\phi_f}$.

3.1 Feature Generation

Cleaning the raw text of the job-postings includes case folding and the elimination of stop words. URLs are recognized and projected to their host name. Therefore, different URLs of the same company are projected to the same token. We also considered to focus on nouns only [1] but did not find a significant improvement.

From the cleaned documents, the *doc2vec* [4] method is applied. As suggested by the results of Le and Mikolov [4], we apply the Distributed Memory model.

3.2 Creation of User Profiles

3.2.1 Preselection of Job Postings. The aim of the preselection is to reduce the number of job postings D to a subset $\tilde{S} \subseteq D$, depending on the user's preferences. The reduced set $\tilde{S}^+$ contains documents that the user prioritizes over the remaining documents. Conversely, the set $\tilde{S}^-$ contains the postings that the user favors least.

The preselection is carried out on the basis of clusters. For this purpose, the job postings $d \in D$ are clustered with the method Mini-Batch k-Means [10]. The clustering creates ϕ_k clusters $C_1, C_2, \ldots, C_{\phi_k}$. Then the clusters are ranked for a user $\tilde{u} \in \tilde{U}$ depending on the number of clicks v on the document $\tilde{d}$ with $(\tilde{d}, v) \in \tilde{L}$ within the corresponding cluster.

The clicks of the users are used as rating. The sigmoid feedback weighting function in Equation 1, ensures that e.g. two clicks on a single document are valued much higher than a single click, while the difference between five and four clicks is not valued very high.

$$\mathrm{fbw}(v) = \frac{v}{1 + |v|} \tag{1}$$

For each cluster center c_i with $i = 1, \ldots, \phi_k$, the cluster weighting w_i sums up the similarities between the clicked job postings $d \in D$ of the user $\tilde{u}$ and the corresponding cluster center c_i:

$$w_i = \sum_{(\tilde{d}, v) \in \tilde{L}} \mathrm{fbw}(v) \cdot \mathrm{sim}(d, c_i) \tag{2}$$

As in the works of Zhang and Cheng [13] and Huang [3], the cosine similarity **sim** is used to compare the document vectors. Based on the values of all clusters, they can be sorted according to the user's preferences. To create the set $\tilde{S}^+$ or $\tilde{S}^-$, the job postings from the highest-ranked or lowest-ranked clusters are assigned to the preselected set up to a desired selection size ϕ_s. This selection size ϕ_s is the matter of evaluation in Section 4.1.

3.2.2 Recommendation Based on KNN Algorithms. The first approach uses the averaged KNN algorithm, in which the profile is formed from the average of all rated job postings of a user. This is a disadvantage for users who have visited job postings from different areas and whose vector therefore lies between these areas. This may cause the vector to point to another area that does not interest the user at all. The idea is to eliminate such areas by preselection.

The according calculation of the user vector $u \in U$ is shown in Equation 3, where the user's rating $\tilde{L}^+$ is applied on the preselected documents $\tilde{S}^+$.

$$u = \frac{1}{|\tilde{L}^+ \cap \tilde{S}^+|} \sum_{(\tilde{d}, v) \in \tilde{L}^+ \cap \tilde{S}^+} \mathrm{fbw}(v) d \tag{3}$$

The set of recommendations $\tilde{R}$ for the user $\tilde{u}$ contains all documents $\tilde{d}_i, \tilde{d}_j \in \tilde{S}^+$, ranked from most similar to least similar by the condition in Equation 4.

$$\forall (\tilde{d}_i, \tilde{d}_j) \in \tilde{R} : \mathrm{sim}(u, d_i) \geq \mathrm{sim}(u, d_j) \tag{4}$$

3.2.3 Recommendation Based on a Hyperplane. The second approach uses the Vuurens et al. [12] method to rank job postings using a hyperplane. This method requires positive and negative rated documents of the user to place the hyperlayer correctly. The preselection is used to determine a negative set $\tilde{L}^-$ for each user. Therefore, a random subset $\tilde{L}^- \subseteq \tilde{S}^-$ is formed, so that $|\tilde{L}^-| = |\tilde{L}^+ \cap \tilde{S}^+|$ applies. In addition, the issue described in Section 3.2.2 occurs where the user may be recommended intermediate job postings from uninteresting areas for rated job postings from different areas.

In each training step, two documents that the user $\tilde{u}$ rated differently, $d_i \in \tilde{L}^-$ and $d_j \in \tilde{L}^+$, are handed over to the neural network. The two documents are then projected with u to a value r_i or r_j, which represents the user's preference $\tilde{u}$. The difference between the two values is then passed to the sigmoid function, which returns the gradient g. The weight vector can be updated using the gradient:

$$u \leftarrow u + \eta \cdot (gd_j - gd_i) \tag{5}$$

where η is the learning rate. After the hyperplane H is trained, all elements are added to the ordered set $\tilde{R}$. Using the normal vector u of the hyperplane of user $\tilde{u}$, all documents $\tilde{d}_i, \tilde{d}_j \in \tilde{S}^+$ are ranked from most similar to least similar by the condition in Equation 6.

$$\forall (\tilde{d}_i, \tilde{d}_j) \in \tilde{R} : u^{\mathsf{T}} d_i \geq u^{\mathsf{T}} d_j \tag{6}$$

4 EVALUATION

Three different evaluation structures are presented in the following. At first, we perform two offline evaluations on the described data set. The evaluation of the preselection in Section 4.1 serves as the detailed analysis of the parameters of the preselection and compares our approaches against the Huang [3] baseline. To evaluate the recommendation performance, a second evaluation is carried out on partial data sets due to the very long runtime of the baseline methods of Guo et al. [2] as well as Zhang and Cheng [13], with the results presented in Section 4.2. An expert-based evaluation follows in Section 4.3 to analyze the realistic applicability of the proposed approach. Since the aim of this work is to maximize the number of relevant documents among the ten highest ranking job postings, the evaluation measure Precision@10 is chosen. A $k = 10$ is chosen as we found it to be a representative value in our evaluation results. The dataset consists of the users' click data containing the corresponding document ID and time stamp for each click (visit) of a job posting. The dataset for evaluation contains approximately 8 million clicks of little more than 300,000 users on almost 2 million job postings. It was analyzed and cleaned as described in the following sections.

4.1 Evaluation of the Preselection

In the following, we present the evaluation of the preselection. Focusing on the evaluation of the selection size ϕ_s, the results of the approaches are compared with the baseline of Huang [3].

For the evaluation of the preselection, we generate a representative sample of users. The *doc2vec* model depends on the size of the feature vector ϕ_f. During evaluation, a feature vector size $\phi_f = 300$ was found to perform best.

Depending on the window size ϕ_w of the *doc2vec* method, we evaluated different part-of-speech filters, focusing on only nouns, on only verbs, and on a combination of these filters. However, the evaluation did not show a significant performance increase for any of the applied filters on our data. The performance on our corpus was found to be optimal for $\phi_w = 10$, which relates to the results of Le and Mikolov [4] who find it to be optimal for a window size of 8.

Evaluating the cluster sizes of 50 clusters up to 250 clusters in steps of 50, we fixed the cluster size for the evaluation at a value of $\phi_k = 100$. The results of the evaluation of the preselection described in Section 3.2 are presented below, by focusing on the selection size ϕ_s for the preselection. We evaluate the selection sizes 50,000 to 600,000 in 50,000-steps, which can be seen in Figure 1, showing the precision of the three methods. The D2V-AVGKNN baseline is constant as there is no preselection applied. Our PRS-D2V-AVGKNN method delivers better results than the baseline (0.0062) as soon as the selection size exceeds 300,000 documents or 15%. The maximum Precision@10 of 0.0073 for our PRS-D2V-AVGKNN approach is reached at 450,000 documents, which corresponds to about 23% of the total document volume. PRS-D2V-HP consistently achieves the worst results in these tests. We think that the reason for this can be found in the negative document set $\tilde{S}^-$, which the procedure requires for placing the hyperplane. As the PRS-D2V-AVGKNN provides very good results, we are certain, that our preselection of positive documents leaves enough variability to cover most relevant documents. On the other hand, the chosen documents for the negative sample set $\tilde{S}^-$ may contribute a too strong claim against possibly relevant documents for the PRS-D2V-HP.

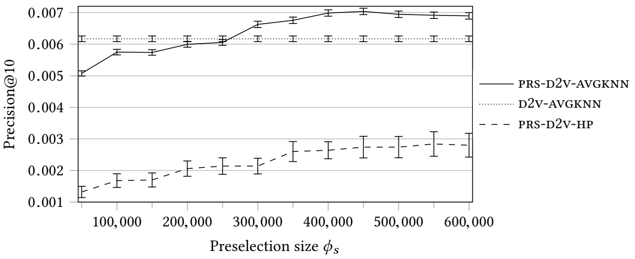

Figure 1: Results depending on the preselection size.

4.2 Evaluation of the Recommendation

We evaluate the performance of the approaches based on the preselection, against the approaches without preselection. Three different subsets of the data with each 1,000 users and 40,000 job postings are sampled.

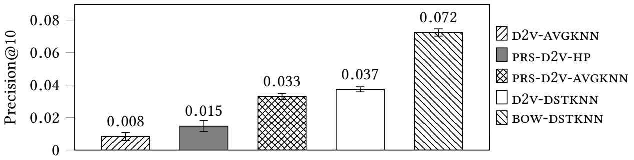

Figure 2: Evaluation results for recommendation.

A look at Figure 2 shows that the baselines BOW-DSTKNN (0.072) and D2V-DSTKNN (0.037) deliver the best results in terms of Precision@10. The baselines BOW-DSTKNN and D2V-DSTKNN both use the distributed KNN algorithm.

When comparing the baselines D2V-DSTKNN (0.037) and D2V-AVGKNN (0.008), which are trained on the same document vectors, it can be observed that the distributed KNN shows a considerably better precision compared to the average KNN. Our PRS-D2V-AVGKNN introduces the preselection to the averaged KNN method, and is able to increase the performance significantly from (0.008) for the D2V-AVGKNN to (0.033) for the PRS-D2V-AVGKNN and almost reaches the performance of the D2V-DSTKNN baseline (0.037). However, the runtime of our preselected averaged KNN approach (PRS-D2V-AVGKNN) stays significantly below the runtime of the distributed KNN (D2V-DSTKNN) baseline.

4.3 Evaluation by Experts

In the offline evaluations it can be seen that in general a very low precision is achieved, as the implicit feedback does not provide a good standard to evaluate against.

Therefore, the PRS-D2V-AVGKNN approach is evaluated using an expert-based evaluation to assess the applicability of the methods in reality. For this purpose, a random sample of 100 users is selected, which are to be examined by three experts from the providing search engine of the data set.

The evaluation shows that about 55% of ten predicted documents represent a meaningful recommendation for users. With a recommendation size of five documents, 58% of the job postings are suitable for the user and with a recommendation size of one document even 60%. It can be concluded that the method presented in this paper can be used to determine appropriate recommendations based on the user's personal preferences.

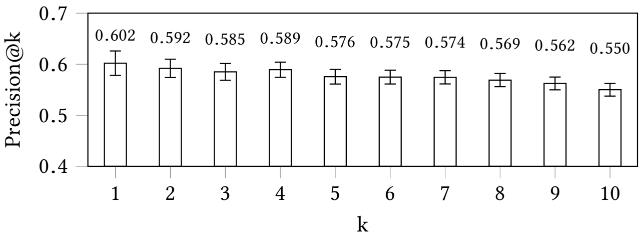

Figure 3: Expert evaluation for PRS-D2V-AVGKNN.

5 CONCLUSION AND OUTLOOK

In this paper, a personalized recommendation system for job postings based on the click data of the users and the textual description of the job postings is presented and evaluated. On the foundation of the *doc2vec* method of Le and Mikolov [4] applied to a corpus of approximately 2 million job postings, a novel approach for the preselection of documents is presented. The presented approach is shown to increase the recommendation performance of a baseline approach Huang [3], while reducing its runtime. On the other hand, the presented approach enables a recommendation approach based on a hyperplane [12]. It was shown that for the Precision@10, our KNN recommender almost reaches the performance of the identified baseline of Zhang and Cheng [13], while staying below the baseline of Guo et al. [2]. However, the runtime of these two baselines (using distributed KNN) increases quadratically with the number of documents, in contrast to our approach (using averaged KNN), which remains as the best applicable approach in the provided scenario.

ACKNOWLEDGEMENTS

This work is supported by Deutsche Forschungsgemeinschaft (DFG) under Grants STE 866/9-2 RE 2593/3-2. The authors alone are responsible for the content.

REFERENCES

[1] Wael Alkhatib, Christoph Rensing, and Johannes Silberbauer. 2017. Multi-label Text Classification Using Semantic Features and Dimensionality Reduction with Autoencoders. In *International Conference on Language, Data and Knowledge.* Springer, 380–394.

[2] Xingsheng Guo, Houssem Jerbi, and Michael P. O'Mahony. 2014. An Analysis Framework for Content-based Job Recommendation. In *22nd International Conference on Case-Based Reasoning (ICCBR), Cork, Ireland.*

[3] Yanbo Huang. 2016. *Exploiting Embedding in Content-Based Recommender systems.* Master's thesis. TU Delft.

[4] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning.* 1188–1196.

[5] Prem Melville, Raymond J. Mooney, and Ramadass Nagarajan. 2002. Content-boosted collaborative filtering for improved recommendations. *Aaai/iaai* 23 (2002), 187–192.

[6] Cataldo Musto, Giovanni Semeraro, Marco de Gemmis, and Pasquale Lops. 2016. Learning word embeddings from wikipedia for content-based recommender systems. In *European Conference on Information Retrieval.* Springer, 729–734.

[7] Marc Poch, Núria Bel, Sergio Espeja, and Felipe Navio. 2014. Ranking Job Offers for Candidates: learning hidden knowledge from Big Data.. In *LREC.* 2076–2082.

[8] Sebastian Schmidt, Steffen Schnitzer, and Christoph Rensing. 2016. Text Classification Based Filters for a Domain-Specific Search Engine. *Computers in Industry* 78 (May 2016), 70–79. https://doi.org/10.1016/j.compind.2015.10.004

[9] Steffen Schnitzer, Sebastian Schmidt, Christoph Rensing, and Bettina Harriehausen-Mühlbauer. 2014. Combining active and ensemble learning for efficient classification of web documents. *Polibits* 49 (2014), 39–45. http://www.scielo.org.mx/pdf/poli/n49/n49a5.pdf

[10] David Sculley. 2010. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web.* ACM, 1177–1178.

[11] Pooja Tripathi, Ruchi Agarwal, and Tanushi Vashishtha. 2016. Review of job recommender system using big data analytics. In *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on.* IEEE, 3773–3777.

[12] Jeroen BP Vuurens, Martha Larson, and Arjen P. de Vries. 2016. Exploring deep space: Learning personalized ranking in a semantic space. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems.* ACM, 23–28.

[13] Chenrui Zhang and Xueqi Cheng. 2016. An ensemble method for job recommender systems. In *Proceedings of the Recommender Systems Challenge.* ACM.