Beyond Accuracy Optimization: On the Value of Item Embeddings for Student Job Recommendations

Emanuel Lacic
Know-Center Graz
Graz, Austria
elacic@know-center.at

Dominik Kowald
Know-Center Graz
Graz, Austria
dkowald@know-center.at

Markus Reiter-Haas
Moshbit GmbH
Graz, Austria
markus.reiter-haas@studo.co

Valentin Slawicek
Moshbit GmbH
Graz, Austria
valentin.slawicek@studo.co

Elisabeth Lex
Graz University of Technology
Graz, Austria
elisabeth.lex@tugraz.at

ABSTRACT

In this work, we address the problem of recommending jobs to university students. For this, we explore the impact of using item embeddings for a content-based job recommendation system. Furthermore, we utilize a model from human memory theory to integrate the factors of frequency and recency of job posting interactions for combining item embeddings. We evaluate our job recommendation system on a dataset of the Austrian student job portal Studo using prediction accuracy, diversity as well as adapted novelty, which is introduced in this work. We find that utilizing frequency and recency of interactions with job postings for combining item embeddings results in a robust model with respect to accuracy and diversity, but also provides the best adapted novelty results.

KEYWORDS

Item Embeddings; Job Recommendations; Novelty; Diversity; Frequency; Recency; BLL Equation

1 INTRODUCTION

Finding jobs is not a trivial task for recently graduated university students since they usually only have a little or no relevant work experience at all. Nowadays, this is a real issue as having a university degree does not automatically guarantee students to get their desired job after graduation. For example, a recent study [4] reports that in 2013, 23.5% of employed U.S. graduates are not only underemployed but also work in positions with a below-than-average salary. We recently started to tackle the same problem for Austrian university students within the Studo mobile application[1]. Studo provides guidance and support for about one third of Austrian students by offering services such as finding the right job[2] in order to gather relevant work experience before they graduate (see Figure 1a).

[1]https://studo.co/
[2]https://jobs.studo.co/

IFUP Workshop @ ACM WSDM '18, February 6, 2018, Los Angeles, USA

DOI: N/A

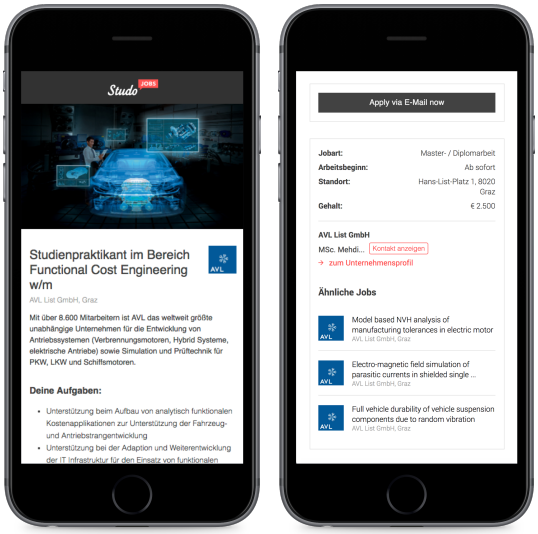

Figure 1: Screenshot of the Studo mobile application, which shows (a) the detail view of a job posting and, (b) a recommendation with low diversity.

In Studo, we currently use a content-based job recommendation system to suggest similar jobs to the currently viewed one. Due to limited display space in mobile applications, we focus on a low number of job recommendations (e.g., 3 or 6) that are shown to the user. As such, we frequently encounter the problem of a poor recommendation diversity. This is also demonstrated in Figure 1b, in which companies often reuse a large portion of their job description text for other open positions, which leads to content-based recommendations that the users may perceive as unhelpful. Apart from recommendation diversity, novelty is also an important metric for personalized job recommendations since applying to popular jobs may decrease the student's satisfaction due to high competition and less chance of getting hired (see e.g., [5]).

The present work. In this work, we apply the Doc2Vec algorithm [8] to obtain fixed-length job vectors to improve recommendation diversity and novelty. That is, we learn neural network-based item embeddings in an unsupervised manner from the job description. This type of item embeddings enables to represent the job description

as a dense vector that takes into account its semantic and syntactic structure.

In order to construct these vectors, we utilize three approaches. Firstly, we use the vector representation of the job posting the user has most recently interacted with. Secondly, we synthetically create a vector representations of all jobs the user has interacted with in the past by averaging the column values of these vectors. Thirdly, we use the Base-Level Learning (BLL) equation from human memory theory [2] to integrate the factors of how frequently and recently the user has interacted with job postings in the past. We evaluate these approaches using a dataset gathered from the job portal used by Austrian students within the Studo mobile application.

Contributions. Our contributions are three-fold: (i) we explore the impact of item embeddings on not only accuracy but also diversity and novelty for recommending jobs to students, (ii) we utilize the BLL equation from human memory theory to model the frequency and recency of job interactions, and (iii) we propose an adapted novelty measure to evaluate job recommendations.

In summary, this study may help researchers to obtain an overview of how to combine learned item embeddings in order to optimize diversity and novelty in addition to accuracy. Our results show that utilizing the BLL equation can improve recommendation performance. Moreover, combining the BLL-induced item embeddings with user-based Collaborative Filtering produces a robust model that gives the best results with respect to our adapted novelty metric.

2 RELATED WORK

Research on job recommmendations has mostly focused on improving the accuracy by applying various Collaborative- and Content-Based Filtering approaches or their hybrid combinations [1, 18]. [10] inverted the problem and searched for the right users for a given job and then recommended this job to the users. Classic machine learning-based methods have also been used for job recommendations. For example, the work of [13] recommended jobs based on inference from the job transition patterns observed in the past. This, however, is not directly applicable for student job recommendations as students often have no job experience at all.

Models that train item embeddings, such as Word2Vec [11], have been shown to provide good results in many natural language processing tasks. As such, research on improving the recommendation performance by employing word embedding vectors is gaining more momentum. As an example, the authors of [12] used the learned word embeddings trained on Wikipedia for content-based recommender systems. Additionally, in [3], word embeddings were employed to represent job postings. In order to achieve a high accuracy, a large number of embedding dimensions were required.

The authors of [17] utilized item embeddings on top of Collaborative Filtering to tackle the cold-start problem for job recommendations. In addition to that, in the RecSys Challenge of 2016, [9] suggested that time-dependent models can enhance the job recommendation performance, which further motivates the applicability of our BLL-based approach.

3 APPROACH

Over the last four years, there has been a number of notable publications in the area of applying deep learning for recommender systems. Especially methods for finding good embeddings (also known as latent representations) for items have become popular. Most models use some variation of Word2Vec [11]. In our work, we utilize the DBOW approach of [8] (also known as Doc2Vec or Paragraph Vector), which works in the same way as skip-gram, except that the input is replaced by a token representing the job posting. Thus, we first train our model using the description text of job postings and afterwards extract the corresponding vector representations.

To train a Doc2Vec model, we need to set several parameters beforehand. These are the window size (i.e., 5, 10, 15 or 20), number of negative samples (i.e., 0, 5, 10, 15) and number of epochs to train (i.e., from 1 to 5). Based on multiple parameter combination experiments, we trained the final item representations using the job description text with a window size of 20, a learning rate of 0.025, 10 negative samples and 1 training epoch.

Recommendation strategy. In order to use the extracted vector representations (i.e., embeddings) in a content-based manner, we need to construct a reference vector for which we can find top-k similar vectors (i.e., job postings) using the Cosine similarity. To obtain this reference vector, we initially define two different strategies:

- LAST We use the vector representation of the job posting the user has most recently interacted with. This strategy simulates the current use-case in Studo of recommending similar job postings to the currently viewed one.
- AVG We use the user's whole job interaction history and synthetically create a reference vector. This is done by averaging the columns of all the vectors that are assigned to the job postings contained in the history.

Integrating frequency and recency of job interactions. Research on how humans are accessing information in their long-term memory has shown that past interaction frequency and recency are crucial factors [2]. In this respect, the Base-Level Learning (BLL) equation, defined by the cognitive architecture ACT-R, integrates these two factors in order to model the information access in human memory. Our previous research has shown that the BLL equation can be utilized for developing social tag and hashtag recommender systems [6]. One issue of our proposed LAST and AVG approaches is that they either focus on the factors of interaction recency (i.e., LAST) or interaction frequency (i.e., AVG) and thus, do not account for both factors simultaneously. Thus, we use the BLL equation to combine the factors of how frequently and recently the user has interacted with job postings in the past:

$$BLL_j = \ln\left(\sum_{i=1}^{n} (TS_{ref} - TS_{j,i})^{-d}\right) \quad (1)$$

Here, BLL_j is the BLL value for a given user u and a given job j, and n states the number of times u has interacted with j in the past. Moreover, $TS_{j,i}$ is the timestamp (in seconds) of when u has interacted with j for the ith time and TS_{ref} is a reference timestamp such as the time when the job recommendations are requested. d is a parameter to set the time-dependent decay of item exposure in human memory and, according to [2], we set it to its default value of 0.5.

These BLL values are first normalized using a softmax function and then multiplied with the individual job vectors from the user's history. This way, we form a weighted sum of job postings based on how frequently and recently the user has interacted with them.

4 METHODOLOGY

To evaluate the performance of our approach, we followed an offline evaluation methodology.

Dataset. We extracted our data from Studo and split it in two different sets (i.e., training and test set) using a method similar to the one described in [7]. Thus, for each user with at least 11 items in the history, we withheld 10 viewed job postings from the training set and added them to the test set to be predicted. This results in 3,011 users who get recommendations from 2,345 job postings. The interaction data contains 140,411 job views, which results in a sparsity of 98.01%.

Experimental setup. We utilize three well-known baselines:

- MP The Most Popular approach recommends for any user the same set of items, which are weighted and ranked by the frequency of job views.
- CBF Content-Based Filtering analyses item metadata to identify other items that could be of interest for a specific user. This is done using TF-IDF on the description text of the job the user has most recently interacted with.
- CF User-Based Collaborative Filtering is defined as recommending jobs to a target user that have been previously viewed by similar users (i.e., the neighbors). To find the k nearest neighbors, we utilize the Cosine similarity metric.

We did not evaluate Matrix Factorization approaches as our focus lies on calculating recommendations in real-time. That is, we do not employ an offline model update strategy that could last hours or days and potentially miss the user's real-time demand. Extracting item embeddings does not present an issue here as this is performed as soon as a new job is posted and thus, can already be recommended.

To quantify the performance of each of our recommender approaches, we use well-established metrics in recommender systems. Furthermore, we propose to evaluate job recommendations with respect to an adapted novelty measure.

Measuring novelty. Recommendation novelty can be seen as the ability of a recommender to introduce users to items that have not been (frequently) experienced before. A recommendation that is accurate but not novel will include items that the user enjoys, but (probably) already knows of. Optimizing on novelty has been shown to have a positive, trust-building impact on user satisfaction [15]. In our experiments, we measure novelty with a normalized metric previously introduced by [19]:

$$Novelty@k = 1 - \frac{1}{|U|} \sum_{u \in U} \frac{1}{k} \sum_{i \in k} \frac{\log_2(pop_i + 1)}{\log_2(pop_{MAX} + 1)} \quad (2)$$

This way we quantify the average information content of job recommendations, where higher values mean that more globally "unexplored" items are being recommended. If the likelihood that a user has experienced an item is proportional to its global popularity, this can serve an approximation of true novelty.

As already mentioned, we aim to optimize the diversity and novelty of recommendations in order to improve long-term user satisfaction. One issue, however, is the over-optimization of the novelty metric as this basically leads to recommending only items from the far end of the long-tail. As such, we investigated the difference in novelty on job postings that are being viewed by the students to the ones the students have actually applied to. We found that the novelty value of the applied jobs N_A is on average 0.58. Thus, we propose to evaluate job recommendation novelty with respect to an adapted novelty measure that considers the desired outcome in the live system. Our goal is to minimize the distance between the calculated recommendation novelty and the average novelty of job applications N_A (i.e., 0.58), which is given by:

$$Novelty^*@k = 1 - |N_A - Novelty@k| \quad (3)$$

Measuring diversity. As defined in [16], recommendation diversity is calculated as the average dissimilarity of all pairs of items in the list of recommended items. Given a distance function $d(i, j)$ that is the dissimilarity between two items i and j, the diversity is given as the average dissimilarity of all pairs of items in the list of recommended items [16]. In our experiments, we use Cosine similarity to measure the dissimilarity of two job postings using the textual content from their descriptions.

Measuring accuracy. nDCG is a ranking-dependent metric that not only measures how many items are correctly predicted but also takes the position of the items in the recommended list into account [14]. It is calculated by dividing the DCG of the user's recommendations with the ideal DCG value, which is the highest possible DCG value that can be achieved if all the relevant items would be recommended in the correct order.

5 RESULTS

In Studo, we show job recommendations via the mobile app, which has limited screen space. As such, in Table 1, we report the evaluation results for $k = 3$ and $k = 6$ recommended jobs (although the recommendation performance for higher k values follows a similar trend).

We found that both MP and CF perform well with respect to accuracy and diversity but suffer from poor novelty scores, which are far from our target novelty $N_A = 0.58$. CBF had opposing results to these, thus suffering from poor accuracy and a rather low diversity. This leads to the previously mentioned situation, in which a user is confronted with very similar job recommendations, which can be perceived as not very helpful. While CBF has a very high novelty score (e.g., the highest one for $k = 6$), it even overshoots the target novelty N_A that we want to reach.

Impact of embeddings for job recommendation. By utilizing item embeddings using the LAST strategy, the accuracy and diversity are slightly improved over CBF. The novelty, on the other hand, is lowered, which is actually desirable as we want to minimize the distance to the target novelty N_A. While utilizing the AVG strategy, we trade the accuracy for the highest diversity results. Increasing the vector dimensions (i.e., 200 and 300), however, leads to higher novelty, which corresponds to a bad performance of the adapted

Approach			k = 3				k = 6			
			Novelty*	Novelty	Diversity	nDCG	Novelty*	Novelty	Diversity	nDCG
MP			.5849	.1649	.7261	.0395	.6057	.1857	.7156	.0722
CBF			.8124	.7676	.4536	.0122	.7965	.7835	.4854	.0156
CF			.7718	.3518	.6736	.0889	.7860	.3660	.6814	.1292
Doc2Vec	LAST	d=100	.8331	.7469	.4845	.0170	.8161	.7639	.5486	.0217
		d=200	.8411	.7389	.5091	.0182	.8212	.7588	.5854	.0219
		d=300	.8448	.7352	.5163	.0177	.8206	.7594	.5953	.0220
	AVG	d=100	.8275	.7525	.6929	.0107	.8144	.7656	.7239	.0154
		d=200	.7930	.7870	.7455	.0099	.8050	.7750	.7830	.0135
		d=300	.7715	.8085	.7439	.0091	.8003	.7797	.7796	.0133
	BLL	d=100	.8500	.7300	.5974	.0156	.8322	.7478	.6486	.0198
		d=200	.8284	.7516	.6408	.0146	.8275	.7525	.7006	.0188
		d=300	.8191	.7609	.6388	.0144	.8222	.7578	.7015	.0186
CF + BLL			.8731	.4531	.6820	.0721	.9578	.5378	.6890	.0900

Table 1: Evaluation results which show that a recommendation approach that uses the BLL equation provides a good balance between accuracy and diversity while achieving the best performance with respect to the target novelty (i.e., the Novelty* measure).

Novelty* score. These results suggest that a combination of LAST and AVG by means of BLL is needed to reach a better balance of accuracy, diversity and novelty.

Effect of frequency and recency. By utilizing the BLL equation when constructing item embeddings, we account for the factors of how frequently and recently the user has interacted with job postings in the past. We can see that the accuracy results are only slightly lower than the ones of the LAST approach. The same is true regarding the diversity when compared to the AVG approach. The BLL approach still outperforms CBF with respect to both metrics. When looking at the target novelty N_A, utilizing the BLL equation results in achieving the best Novelty* performance and this even with the smallest number of vector dimensions (which is even more favorable when similarity calculations are performed in real-time). Finally, by combining the best performing BLL approach with Collaborative Filtering in a round robin fashion, we can recommend jobs with nearly the target novelty N_A (i.e., 0.58, while still providing high accuracy and diversity scores.

6 CONCLUSION AND FUTURE WORK

In this paper, we explored the impact of item embeddings for tackling the problem of recommending jobs to university students. We proposed how to integrate the factors of frequency and recency of job interactions when combining item embeddings for content-based recommendations. Our evaluation conducted on a dataset gathered from the Austrian student job portal Studo showed that a recommendation approach that uses the BLL equation from human memory theory provides a good trade-off between accuracy and diversity while achieving the best performance with respect to our adapted novelty measure. Moreover, combining the BLL approach with Collaborative Filtering can lead to a robust model with respect to accuracy, diversity and novelty.

For future work, we plan to perform an online study to evaluate the real user acceptance of utilizing the BLL equation for constructing item embeddings. Furthermore, we plan to investigate how to integrate university-related metadata (e.g., recently visited lectures, expected date of graduation, etc.) within the trained item embeddings in order to further enhance our job recommendation system.

REFERENCES

[1] S. T. Al-Otaibi and M. Ykhlef. A survey of job recommender systems. *International Journal of Physical Sciences*, 7(29):5127–5142, 2012.
[2] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. An integrated theory of the mind. *Psychological review*, 111(4):1036, 2004.
[3] Y. Huang. Exploiting embedding in content-based recommender systems. 2016.
[4] J. Jones, J. Schmitt, et al. A college degree is no guarantee. Technical report, Center for Economic and Policy Research (CEPR), 2014.
[5] K. Kenthapadi, B. Le, and G. Venkataraman. Personalized job recommendation system at linkedin: Practical challenges and lessons learned. In *Proc. of ACM RecSys'17*.
[6] D. Kowald, S. C. Pujari, and E. Lex. Temporal effects on hashtag reuse in twitter: A cognitive-inspired hashtag recommendation approach. In *Proc. of WWW'17*.
[7] E. Lacic, D. Kowald, and E. Lex. Tailoring recommendations for a multi-domain environment. In *Proc. of RecSysKTL'17 co-located with ACM RecSys'17*.
[8] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proc. of ICML'14*.
[9] K. Liu, X. Shi, A. Kumar, L. Zhu, and P. Natarajan. Temporal learning and sequence modeling for a job recommender system. In *Proc. of RecSys Challenge'16*.
[10] R. Liu, W. Rong, Y. Ouyang, and Z. Xiong. A hierarchical similarity based job recommendation service framework for university students. *Frontiers of Computer Science*, 11(5):912–922, Oct 2017.
[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
[12] C. Musto, G. Semeraro, M. de Gemmis, and P. Lops. Learning word embeddings from wikipedia for content-based recommender systems. In *Proc. of ECIR'16*.
[13] I. Paparrizos, B. B. Cambazoglu, and A. Gionis. Machine learned job recommendation. In *Proc. of ACM RecSys'11*, New York, NY, USA.
[14] D. Parra and S. Sahebi. Recommender systems : Sources of knowledge and evaluation metrics. In *Advanced Techniques in Web Intelligence-2: Web User Browsing Behaviour and Preference Analysis*, pages 149–175. Springer, 2013.
[15] P. Pu, L. Chen, and R. Hu. A user-centric evaluation framework for recommender systems. In *Proc. of ACM RecSys'11*.
[16] B. Smyth and P. McClave. Similarity vs. diversity. In *Proc. of ICCBR '01*.
[17] J. Yuan, W. Shalaby, M. Korayem, D. Lin, K. AlJadda, and J. Luo. Solving cold-start problem in large-scale recommendation engines: A deep learning approach. In *Proc. of Big Data'16*. IEEE.
[18] C. Zhang and X. Cheng. An ensemble method for job recommender systems. In *Proceedings of the Recommender Systems Challenge*, page 2. ACM, 2016.
[19] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515, 2010.