UDC – 681.5:62.868.8

DEVELOPING BI-DIRECTIONAL RECOMMENDATION SYSTEM BASED ON CONTENT FOR JOBS AND MATCHING CANDIDATES USING FAST TEXT MODEL

Narek H. Hovsepyan
Russian-Armenian University
123 Hovsep Emin St., Yerevan 0051, RA
narek.hovsepyan@student.rau.am
ORCID iD: 0000-0003-1615-4963
Republic of Armenia

Abstract

In this paper, a content-based bi-directional candidate-job recommender system is suggested for matching the best candidates for a given job and vice versa. Our solution is based on deep learning techniques. It uses applicant information (title, skills, location, etc.) and job descriptions/information to find the best suggestion for a given job or candidate. For evaluation of the suggested recommendation system the Job2Candidate dataset has been constructed. It is shown that using the FastText model and seniority level allows our approach to get more accurate results than other existing alternative approaches give.

Key words: recommendation system, candidate recommendation, job recommendation, NLP, FastText, language models, unsupervised learning.

Introduction

Recommender systems are being utilized in a wide range of applications. However, depending on the domain in which it is used, the type of advice offered may vary. We need to discover a way to rank all candidates from a certain pool for a specific position to create a candidate recommendation system based on the job description. The candidate recommender will assist recruiters in locating and contacting the finest prospects. As it is bi-directional, the solution can also be used to suggest jobs for specified candidates. However, the candidate recommendation system is our primary focus.

Most candidate and job recommendation systems [1, 2, 4, 6] utilize collaborative filtering (CF) [7, 8, 9], and just a few methods [3, 5] contain some content-based techniques which are only used to overcome the so-called cold-start problem of CF approaches and they are all strongly CF-based approaches in general.

Our system is unique as it focuses solely on content-based suggestions outperforming previous alternatives.

Conflict setting

The aim of the work is to develop a recommender system for jobs to make efficient the process of finding the best options from a certain pool of candidates.

Datasets

The main dataset which we have used for training our model is the jobs dataset. Also, we have constructed the Job2Candidate dataset, which mainly helps in the evaluation stage.

Jobs dataset. We have crawled the most popular job posting websites to create a jobs dataset, which contains job titles and descriptions from various industries. Crawled and the filtered dataset contains over 15 million job titles and descriptions. All titles and descriptions are written in English. All the job titles and descriptions are concatenated to form a big text corpus which was used to train our model.

Job2Candidate dataset. We gathered a job-candidate dataset, which contains the title and description for the job, and the title, skills and bio for the candidate (which also includes previous experience). Using job and candidates sets, we have manually created the Job2Candidate dataset, where for each pair of jobs and candidates, we have label 1 if the candidate is a good enough match for suggesting and 0 otherwise. Then we combined it with our internal “offers and hires” dataset which contains job and applicant pairs where the candidate was hired or received an offer for that job. As a result, the dataset is evenly balanced with almost 3000 job-candidate pairs with labels of 0 or 1. We utilized this dataset to evaluate a solution's capacity to filter out bad candidates and recommend only those who are a good enough match.

Models and Methods

We have used the FastText [10] model and jobs dataset to get our model. FastText is a commonly used model for word embedding. It is an extension of word2vec [11], created by Facebook. It uses a fast and effective method to learn word representations and perform text classification. The model was used to get embeddings for both candidate information and job information, then cosine similarity of two embeddings gives a score between 1 in case the job and the candidate are a perfect match and -1 otherwise. The same cosine similarity score was used to rank suggested candidates or jobs.

We have used the standard FastText model to help us get embeddings for our job and candidate information. Choosing the FastText over the standard word2vec algorithm is based on the FastText extra feature to provide embeddings even for tokens that are out of train vocabulary using some techniques with ngrams. The FastText model works pretty fast both in the training and the evaluation phase.

With the FastText model, we are using job and candidate information separately to get their embeddings. We are using only professional information for embedding, which means no personal information is used in the recommendation system, which helps avoid biases. We use the same model for jobs and candidates, which means all embeddings are from the same distribution space and have the exact dimensions. For each job and candidate pair, we are calculating the cosine similarity using their embeddings.

$$similarity = \frac{\sum_{i=1}^{N} A_i B_i}{\sqrt{\sum_{i=1}^{N} A_i^2} \cdot \sqrt{\sum_{i=1}^{N} B_i^2}}, \ where\ A_i, B_i \in R^N \quad (1)$$

The following formula is used to calculate the score, which is bounded between -1 and 1. N is the dimension of embeddings. As a result, each job candidate pair has a score. We have used a threshold of 0.54 to filter candidates. All candidates with scores below 0.54 are marked as not relevant candidates. The rest of the candidates are ordered via their respective scores. The grid-search algorithm found the given 0.54 threshold.

The technique mentioned above works well, but there is a problem with seniority levels. For example, suppose we need a Senior javascript engineer with skills [javascript, jquery]. In that case, the same terms can be matched in junior developers’ profiles, but we do not have to suggest them. It became natural to include a new seniority level feature to address this problem, which allows us to match candidates by content and seniority level. We used a keyword search in the job title and extracted experience requirements from the job description to get seniority level for the job. The same keyword search method is used for the candidate titles. We combine that with calculating years of experience from candidates' experience to get seniority level for candidates. As stated in Tab. 1, there are six levels of seniority.

Table 1

Seniority levels

Level 0	Interns, juniors, …
Level 1	Mid-level, assistants, …
Level 2	Senior-level, …
Level 3	Managers, team leads, …
Level 4	Directors, senior managers, …
Level 5	C-level executives, owners, founders, …

After detecting the seniority level for candidates, we are filtering out candidates with mismatching seniority levels.

Research results

As previously stated, we evaluated using the Job2Candidate dataset, but this is only the first stage of our system's evaluation. Accuracy, recall, and precision were the metrics we used in the first stage.

We have defined a metric called Top10-referrals for the second stage of evaluation, which is customer feedback/usage. Our platform's referral/hire events are used to calculate the Top10-referrals metric. The Top10-referrals metric indicates how many percent of the customer's referrals/hires came from our top 10 recommendations.

Let us denote each referred/hired candidate with x_i. Define k_i , which is equal to 1 if x_i is in the top 10 suggestions of recommendation, otherwise 0.

The top10-referrals metric is calculated as shown below:

$$y = \frac{\sum_{i=1}^{N} k_i}{N} \tag{2}$$

where N is the number of referred/hired candidates.

Here are the results of several months of testing with the existing algorithms and the currently described technique.

Evaluation stage one results are shown in Tab 2:

Table 2

Evaluation stage one, results

Solution name	Accuracy	Precision	Recall
CF (without cold start solution)	68.91%	61.66%	74.12%
CF + content-based solution	71.32%	67.37%	77.01%
Content-based solutions with Solr	76.01%	75.33%	78.50%
FastText model (without seniority level)	77.01%	79.97%	74.26%
FastText model (with seniority level)	86.66%	81.36%	92.69%

On the first step of evaluation, our solution outperforms previous CF-based alternatives. We deployed our solutions for 100+ customers for several months (10000+ referred/hired candidates) in the Team able platform for the second stage of review, and the results are displayed in Tab 3.

Table 3

Evaluation stage two, results

Solution name	Top10-referrals metric
CF (without cold start solution)	9.20%
CF + content-based solution	10.01%
Content-based solutions with Solr	18.88%
FastText (without seniority level)	28.51%
FastText model (with seniority level)	35.94%

Our approach surpasses other existing alternatives by a large margin during the review process. Furthermore, our approach eliminates the cold start issue, which is a significant issue for new consumers. Also, our solution is bi-directional, which means it can be used as a job suggestions system for the specified candidates.

Conclusion

We introduced an innovative strategy in this research that allowed us to surpass prior CF-based and content-based approaches. It also does not have a cold start issue and performs better with or without previous data. In terms of execution time, our solution is relatively quick, even with over 1 million candidates. The calculation of embeddings and respective scores is a one-time job, after which we only need to use those scores to filter and rank candidates.

The described solution is bi-directional which means it can be used as a job suggestions system for the specified candidates. As of now, we have used the job suggestion part in our product, but there is no automated way of tracking the effectiveness on our platform.

References

1. Koh M.F., Chew Y.Ch. Intelligent job matching with self-learning recommendation engine (2015) //Procedia Manufacturing 3.- p. 1959-1965.
2. Patel R., Vishwakarma S.K. An Efficient Approach for Job Recommendation System Based on Collaborative Filtering (2019) //ICT Systems and Sustainability: Proceedings of ICT4SD 2019, Volume 1, 1077.- p. 169.
3. Belsare R.G., Deshmukh D.V. Employment Recommendation System using Matching, Collaborative Filtering and Content Based Recommendation (2018) //Int. J. Comput. Appl. Technol. Res 7.6.- p. 215-220.
4. Yan X, et al. Social skill validation at linkedin (2019) //Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.- p. 2943-2951.
5. Yang Sh., et al. Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive Statistical Relational Learning approach (2017) //Knowledge-Based Systems 136.- p. 37-45.

6. Shalaby W., et al. Help me find a job: A graph-based approach for job recommendation at scale (2017) //2017 IEEE international conference on big data (big data).- p. 1544-1553.
7. J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang Recommender system application developments: a survey (2015) //Decis. Support Syst. 74.- p. 12-32.
8. Su X., Khoshgoftaar T.M. A survey of collaborative filtering techniques (2009) //Advances in artificial intelligence, 2009.- p.421-425.
9. Schafer J.B., et al. Collaborative filtering recommender systems (2007) //The Adaptive Web. Lecture Notes in Computer Science, vol. 4321, Springer, Berlin, Heidelberg.- p. 291-324.
10. Bojanowski P., et al. Enriching word vectors with subword information (2017) //Transactions of the Association for Computational Linguistics 5.- p. 135-146.
11. Mikolov T., et al. Efficient estimation of word representations in vector space (2013) //arXiv preprint arXiv:1301.3781.

ԲՈՎԱՆԴԱԿՈՒԹՅԱՆ ՎՐԱ ՀԻՄՆՎԱԾ ԵՐԿԿՈՂՄԱՆԻ՝ ԱՇԽԱՏԱՆՔՆԵՐ ԵՎ ԹԵԿՆԱԾՈՒՆԵՐ ԸՆՏՐԵԼՈՒ ԱՌԱՋԱՐԿՈՒԹՅՈՒՆՆԵՐԻ ՀԱՄԱԿԱՐԳԻ ՄՇԱԿՈՒՄԸ՝ ՕԳՏԱԳՈՐԾԵԼՈՎ FAST TEXT ՄՈԴԵԼԸ

Ն.Հ. Հովսեփյան
Հայ-Ռուսական համալսարան

Այս հոդվածում առաջարկվում է բովանդակության վրա հիմնված երկկողմանի թեկնածու-աշխատանք առաջարկող համակարգ՝ տվյալ աշխատանքի համար լավագույն թեկնածուներին գտնելու համար և հակառակը: Մեր լուծումը հիմնված է խորը ուսուցման տեխնիկայի վրա: Այն օգտագործում է դիմողի տեղեկատվությունը (անվանումը, հմտությունները, գտնվելու վայրը և այլն) և աշխատանքի նկարագրությունները/տեղեկությունները տվյալ աշխատանքի կամ թեկնածուի լավագույն առաջարկը գտնելու համար: Առաջարկվող համակարգի գնահատման համար ստեղծվել է Job2Candidate տվյալների հավաքածուն: Ցույց է տրված, որ FastText մոդելի և ավագության մակարդակի օգտագործումը թույլ է տալիս մեր մոտեցմանը ստանալ ավելի ճշգրիտ տվյալներ, քան գոյություն ունեցող այլ մոտեցումների տված արդյունքները:

Բանալի բառեր. առաջարկությունների համակարգ, թեկնածու առաջարկ, աշխատանքի առաջարկ, NLP, FastText, լեզվի մոդելներ, առանց ուսուցչի ուսուցում:

РАЗРАБОТКА СИСТЕМЫ ДВУНАПРАВЛЕННЫХ РЕКОМЕНДАЦИЙ НА ОСНОВЕ СОДЕРЖАНИЯ ДЛЯ ПОДБОРА РАБОТЫ И КАНДИДАТОВ С ИСПОЛЬЗОВАНИЕМ МОДЕЛИ FAST TEXT

Н.Г. Овсепян
Российско-Армянский университет

В данной статье предлагается основанная на содержании двунаправленная рекомендательная система кандидатов на работу для подбора лучших кандидатов на данную должность и наоборот. Наше решение основано на методах глубокого обучения. В нем используется информация о кандидате (должность, навыки, местонахождение и т.д.) и описание/ информация о вакансиях, чтобы найти лучшее предложение для данной работы или кандидата. Для оценки предложенной системы рекомендаций был построен набор данных Job2Candidate. Показано, что использование модели FastText и уровня старшинства позволяет нашему подходу получить более точные данные, чем другие существующие подходы.

Ключевые слова: система рекомендаций, рекомендация кандидата, рекомендация работы, NLP, FastText, языковые модели, обучение без учителя.

Submitted on 05.10.2021.
Sent for review on 06.10.2021.
Guaranteed for printing on 02.11.2021.