Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 122 (2017) 865–872

Procedia Computer Science

www.elsevier.com/locate/procedia

Information Technology and Quantitative Management (ITQM2017)

Topic Modeling Driven Content Based Jobs Recommendation Engine for Recruitment Industry

Shivam Bansal[a], Aman Srivastava[a], Anuja Arora[c]*

[a]Data Science Lead, Prophesee Solution Pvt. Ltd., Delhi, India
[b]*Data Scientist, Prophesee Solution Pvt. Ltd., Delhi, India*
[c]*Department of Computer Science, Jaypee Institute of Information Technology, Noida, India*[a]

Abstract

A number of postings for different job roles and job positions are posted at numerous sources in the recruitment industry. Therefore, this is a challenging and time-consuming task to collate the information and find out most relevant user-job connection mapping according to the skills and preferences of a user. This research work has been done to cover up this same problem and efforts have been made to provide a feasible and efficient solution for the same. We suggest a content-based recommendation engine, which automatically provides best suggestions to users by matching their interests and skills with the features of a job posting. In order to produce an intended recommendation, the proposed engine applies various text filters and feature similarity measurements. Similarity techniques use the bag of n-grams and topic models as the elements of feature vectors. The validations and testing of the model on real data obtained from a top job posting website show the applicability and efficiency of using topic models as features. The approach is generic and can be replicated to different industries.

Peer-review under responsibility of the scientific committee of the 5th International Conference on Information Technology and Quantitative Management, ITQM 2017.

Keywords: Topic Modeling; Information Retrieval; Part-of-Speech Tagging; N-gram; TF-IDF

1. Introduction and Literature Study

Due to a large growth of online data in recent years, the task of extracting the relevant information has become a tedious process for the recruitment industry. This industry is heavily dependent upon accurately estimable information filtering systems. Due to this data boost, job listings and recruitment service websites have experienced a dip in quality and performance. In the current scenario, both job seekers and recruiters have to undergo a colossal pool of job descriptions and job listings information in order to come up with final short-listings. On the other end, this process is completely manual due to which there are high chances of missing out of the appropriate and suitable job or candidate, plus it is also a time-consuming process.

* Anuja Arora. Tel.: +91-9810982939.
*E-mail address:*anuja.arora29@gmail.com

1877-0509
Peer-review under responsibility of the scientific committee of the 5th International Conference on Information Technology and Quantitative Management, ITQM 2017.
10.1016/j.procs.2017.11.448

As per the increasing demand for improved and personalized services for job seekers and recruiters, recommendation engines should be able to get the most suitable information from the available resources. For this purpose, we present a prototype of novel implementation of recommendation algorithm [2] that applies topic modeling technique - Latent Dirichlet Allocation (LDA) [1, 3] on jobs data along with other feature engineering methods to define features of a job posting and user profile. The engine is natural language processing driven, enriched with n-grams and latent topics as features, it is composed of data pre-processing and context similarity techniques. Several techniques such as Levenshtein distance [5, 13], fuzzy and semantic matching is performed on the item's data and user's data features. The similarity score is then computed which is further normalized using min - max transformations. Thus, a user-job similarity matrix is obtained that determines the closeness of a job to a user. Collaborating filtering models are associated with a cold start problem and there are high chances of observing irrelevant recommendations due to seasonality changes and constantly changing user behaviors. The content based job recommendation [16] system mentioned in this paper is thus inspired by the need to use text mining algorithms to complement different actions of recruitment industry - job search, screening and short listing with a constructive mechanism. This not only provides accurate and relevant information filtering but also reduces the taken time and human effort in the whole process. To improve the accuracy of results, we have incorporated latent features of the text using topic modeling algorithms. The LDA algorithm is of particular interest as it can produce higher dimensional topics than any other algorithm. Although, the root of recommendation systems have become fairly early in the history of computing but recommendation research got a boost with the increasing era of social network and e-commerce sites. Recommendation systems have become the necessity for social network websites and have become a mainstream research field. While studying the papers related to this, the research challenges such as information overload, constraint based recommendation; content sparsity, cold start etc. provide novelty to the recommendation systems. As a result of these new developments, recommender systems have been the interesting area of research and mainly all application areas have adopted recommender system. Recruitment websites also require a suitable recommender system to recommend the job information to the uses and provide a relevant user list to recruiter according to the job vacancy. Few researchers worked in this same domain using content based recommendation system to provide an efficient system to recruitment agencies but there is a scope of further improvement using machine learning and natural language processing techniques. Toon et. al. [14] worked in this recruitment domain and proposed a hybrid job recommendation in 2016.

The proposed hybrid job recommendation algorithm is in context of RecSys Challenge 2016 [16] and basically used two approaches – content based and a KNN approach. Even in 2013, Yao Lu et. al.[4] proposed a recommender system for job seekers and recruiters. They applied a hybrid approach which brings content based recommendation and graph based page rank approach together to improve result accuracy and to provide the personalized recommendation. Jochen et. al. [6] matched people and job towards a bilateral person-job recommendation system. They contributed two models, CV-Recommender and Job-Recommender. The proposed CV Recommender is a probabilistic hybrid recommendation engine based on latent features of individual preference and Job recommender recommends jobs to the candidate based on their preference profiles. Item recommendation by topic modeling approach is done by Sang Su Lee et. al. and they employed a modified LDA model which generate clusters by topics and include users and tags in an appropriately formed cluster. Even represent results according to changing user interest. We also have used content based filtering recommendation system and latent feature to help job seekers and recruiters. In content based filtering system, users or item based rating vector are used to compute the similarity between users/ items which in turn uses neighborhood based method [7] to find similarity between users/ items.

Besides offering relevant recommendations to the user, we formulate the goal to overcome the major challenges of content based recommendation algorithm in this research work, challenges are as follows-

- The first major challenge of this content based recommendation approach is the sparsity of user-item rating matrix i.e. many of the entries in the matrix would be NULL as there can be many items which are not rated by the users.
- The second challenge is cold start problem, i.e. if a user is new to the system then preferences are not known to the system which makes the recommendation process less reliable.
- The third challenge is to provide the recommendation with maximum likelihood.

The rest of the paper unfolds as follows. Section 2 discusses the data set and preprocessing performed on that dataset. Section 3 is about topic modeling driven content based recommendations approach which provides an overview of the methods used in our proposed model. Text filtering is applied which is discussed in section 4. Section 5 discusses feature similarity technique and their corresponding experiments performed on taken dataset. Further, generated recommendation results are presented in section 6. Finally, section 7 concludes the paper.

2. Dataset Description and its Pre-processing

The complete data consist of two data sets - user details and jobs data. Both of these datasets are obtained from a top recruitment website of India. User data comprises of demographic variables -name, gender, age, country, skills, and interests. Jobs data comprises of job title, job role, and job descriptions. The detailed breakdown of the datasets is shown in Table-1. As an initial preprocessing step, we removed duplicated rows, empty job descriptions and very small job descriptions with keywords less than five. We also removed the outliers (minority classes in jobs data), which in this case was the user designations with less than 10 records in the entire dataset. A complete corpus of 32,163 job data and11, 244 unique users are obtained.

Table 1. Dataset description

User Data		Job Data	
# of unique users	11,244	# of unique job postings	32,163
# of unique skills	2,946	# of unique job roles	7176
# of unique interests	554	# of unique job designations/categories	351

3. Use of Topic Modeling in Content Based Recommendation Engine

The fundamental of the content-based approach is the features present in the user profile and the job description. In text data, words can directly act as features, however, a better representation of features in the text is topics. Topics are defined as the most significant words used in the text corpus. Using topics as features give an advantage in content based recommendation engines – some of the low occurring terms might seem irrelevant with respect to a job recommendation but these terms might be linked with other high-frequency terms which are the strong features. In this paper, we use topic models using LDA to improve our feature vectors.LDA is a hierarchical Bayesian generative model, in which documents are represented as a mixture of a limited set of topics, where each topic is characterized by a distribution over words. LDA allows the representation of documents in a reduced feature space consisting of K dimensions rather than a larger space that contains as many dimensions as the number of unique words in a given corpus. LDA[1] is thus used as a method of unsupervised machine learning to discover latent topics that are hidden in a text. LDA[1] can be represented by plate notation to describe the generative process wherein external plate represents documents, while the inner plate is basically the iterative choice of topics and words within a document. Figure1 presents the LDA model, which is the fundamental diagram of it [1]. Each document in the corpus is generated by picking a multinomial topic distribution $\theta \sim Dirichlet(\alpha)$ for document m. Each word of the document is assigned a topic from the topic distribution. Given the topic, a word is drawn from the multinomial word distribution $\beta k \sim Dirichlet(\eta)$ from the dictionary for that topic k. W is the topic for nth word in document m and is the specific word. The method used

in our model for approximating the LDA is the online learning algorithm, described by Hoffman et al.[17] which is a variation on the expectation maximization approach. The learning algorithm is streamed and runs in constant memory with regard to the number of documents and can make use of a distributed system, which allows it to be implemented on a much larger corpus. We apply LDA to the job function/job role to find topic level distribution and extract topics out of these job roles on the basis of the user mentioned job roles in his profile. Basically, LDA [18] assigns probability along with most appropriate topic chosen from the job role topic. LDA based profile matching is the best practice because it matches profiles according to its topic category otherwise matching of the job on the basis of a specific skill in profile such as recommend just on the basis of 'R', 'C' or 'Java' and to recommend only its associated jobs i

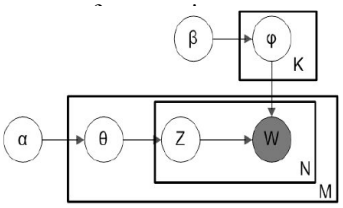


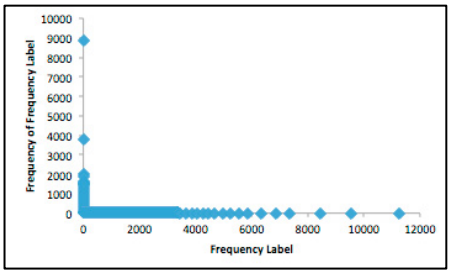
Fig. 1. Basic LDA Model

4. Text filters

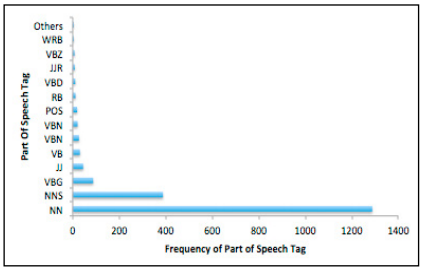
The quality of recommendations is directly dependent upon the features present in the data. However, every feature might not be of equal importance. For example–Few parts of speeches such as pronouns and adjectives do not define job roles. Another case is about the terms which occur very rarely in the entire dataset. By applying relevant filters on the dataset, these weak features can be filtered out and a cleaned corpus can be obtained. Therefore, to obtain useful features and remove ineffective features, we apply frequency filters and part of speech tags (POS) filters in our datasets.

4.1. Frequency Filter:

The corpus contains a total of 748,048 terms of which 21,865 are unique terms. The graph shown in figure 2(a) depicts the word frequency distribution of the data. In the graph, the x-axis represents the word frequency and y axis shows the number of terms having that frequency and shows that the corpus is mostly defined by rarely occurring terms which are the bulkier section as seen in the bottom left. All of these features essentially make the corpus very sparse, hence unstable Frequency filter takes care of weak features and banishes the terms having low frequency. The threshold value is derived from individual documents using the methodology described below.



(a) Word Frequency Distribution Graph



(b) Part of Speech Tag distribution Graph

Fig. 2. Text Filters

The used system contains the corpus D, with m=|D| documents. For each document, word vector is defined as $\{w_1, w_2, w_3 \ldots\ldots, w_n\}$ and number of unique term corresponding to each document are $\{u_1, u_2, u_3 \ldots\ldots, u_n\}$. The average frequency of words in documents is computed as $\left(\frac{w_{D1}}{u_{D1}}, \frac{w_{D2}}{u_{D2}}, \ldots\ldots\ldots\ldots\ldots\ldots\ldots, \frac{w_{Dn}}{u_{Dn}}\right)$. So, Frequency filter takes care of weak features and get rids of terms having low frequency. The threshold value is derived from

individual documents using following formula. Frequency Filter= $min\left(\frac{w_{D1}}{u_{D1}}, \frac{w_{D2}}{u_{D2}}, \ldots\ldots\ldots\ldots\ldots\ldots, \frac{w_{Dn}}{u_{Dn}}\right)$. Suppose document D_1 contains 500 words and 50 unique words, document D_2 contains 500words and 100 unique words and document D_3contains 500 words and 250 unique words. Computed Frequency Filter is $min\left(\frac{500}{50}, \frac{500}{100}, \frac{500}{250}\right) = \min(10, 5, 2)$. So, we remove all the terms with the frequency less than 2.

4.2. Part of Speech Filter:

For the remaining terms, the part of speech distribution is applied. Part of speech distribution graph of the terms present in the corpus is shown in figure 2(b). According to observation, we notice that NN, NNS, VBG, VBN and JJ are highly important as compared to any other POS tag. Also, these are the tags which define the majority of the corpus. In this filter, only important POS tags are selected and all the terms with other tags are removed.

5. Feature Similarity Techniques

In this section, research attempts have been done to implement various feature similarity techniques according to outcome requirements [8]. Instead of choosing one specific feature similarity technique and accepting its result as the final result, different similarity methods have been used [10]. Salton et.al. mentioned in his work that there is not a single universally optimal feature similarity technique [9].Moreover, the ensemble of different similarity matching techniques might give better recommendation approximation. As the concern of best recommendation and similarity measure, we chose to validate our methodology with two well-known similarity measure techniques: flexible string matching [11], cosine similarity measure [12].

5.1. Feature Flexible String Matching

A pivotal element in feature similarity matching is flexible string matching [11] in job data set. Such information is prevalent in our user and job datasets (for e.g. job title and job role in job data; user job title and user skill in user data). To effectively deal with flexible string matching and taking into account of this sort of semi structured data, is a challenge. We now present an experiment of flexible string matching on the considered data set. Jobs table consists of a total of 32,163 rows of job attributes: job role and job designation; user data contain 11,244 rows of user attributes: user skill and user interest which is used for this purpose. We ran a series of flexible string matching techniques for the job data- user data pair. Here, the feature vectors of user and job postings are checked for the closeness of match and we performed exact matching among dimensions of users and jobs. Table 2(a) and table 2(b) shows the sample of job and user feature vector respectively. N represents the feature i.

Table 2(a). Sample job data vector

	N1'	N2'	N3'	N4'	N5'
Job1	ML	R	SQL	NLP	Testing
Job2	Ruby	Content	SQL	SEO	Quality
Job3	Hadoop	DS	Hive	Python	Testing

Table 2(b). Sample user skill data vector

	N1	N2	N3	N4	N5
User1	Python	SQL	NLP	Mongo	DBMS
User2	Oracle	JAVA	XML	SQL	DBMS
User3	Bank	Care	SQL	Audit	Finance

Table 3 shows the sample results of feature flexible similarity matching which yields the similarity user-job matrix.User1 and Job1 show maximum likelihood because of overlapping of skills (NLP, SQL).

Table 3. Sample flexible string matching similarity matrix

	Job1	Job2	Job3
User1	2	0	1
User2	0	1	0
User3	1	1	0

5.2. Feature Vectors Cosine Similarity

The cosine similarity [15] between two vectors is defined as the measure of the normalized projection of one vector over the other. It is defined by the following formula

$$similarity = \cos\theta = \frac{A.B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

It is a measurement of orientation and not the magnitude. It can be seen as a comparison between documents on a normalized space because it does not take the magnitude of each word count of each document into the consideration only the angle between them is considered [15]. Table 4(a) and Table 4 (b) show sample data of number of time (frequency) a specific feature is present in user profile and job profile respectively

Table 4. Sample User Profile Feature Mapping Frequency

	N1	N2	N3	N4	N5	N6	N7	N8	N9
User1	10	7	9	5	0	0	2	0	0
User2	0	5	0	2	9	10	6	0	0
User3	0	8	0	0	0	0	0	9	9

Table 4(b). Sample Job data Feature Mapping Frequency

	N1	N2	N3	N4	N5	N6	N7	N8	N9
Job1	5	6	0	6	0	0	0	0	0
Job2	8	7	10	8	0	0	0	0	0
Job3	0	0	0	0	10	5	8	0	0

Finally, we compute the cosine similarity on User-Job feature mapping data set and get the similarity between user and job corresponding to it. Table 5 shows the sample result of cosine similarity matching. User's job profile - User1 is more similar to job description - Job2 which shows the highest cosine similarity score 0.95 followed by Job1, Job3. Similarly, User2 is more similar Job3.

Table 5. Sample Cosine Similarity user-job similarity matrix

	Job1	Job2	Job3
User1	0.76	0.95	0.09
User2	0.27	0.19	0.86
User3	0.30	0.20	0.07

6. Generated Recommendation Result

6.1. Weighted TF-IDF computation on user data and Job Data

In the user data set, each job category had overlapping frequency count from user's profile for a skill as presented in Table 6 which shows that among the corpus of 11244 user's profile, 521 contains the skill Mongo, 192 contains R and so on.

Table 6. Job Category Vs. User skill Count Matrix

	Mongo	R	SQL	DBMS	HTML	C
Business Analyst	33	24	17	15	15	8
Human Resource	2	21	1	3	1	1
Software Developer	0	3	16	6	31	4
….	…	..	..	….	…	..
Total Frequency	521	192	61	87	91	54

For recommendation, first, we computed weighted Term Frequency (TF) on user data to dampen the effect of high frequency. Weighted TF is computed using TF= $1+\log_{10}$(TF). Where, TF is term frequency of a specific skill in a specific job category in user data as presented in table IX as well. After TF computation on job category and user skill in user data, we will get TF score as some sample shown in table 7.

Table 7. Term frequency of user skill in all job category in user data

	Mongo	R	SQL	DBMS	HTML	C
Business Analyst	1.51	1.38	1.23	1.17	1.17	0.90
Human Resource	0.30	1.32	0	0.47	0	0
Software Developer	0	0.47	1.20	0.77	1.49	0.60
Web Analytics	0.77	0	1	0.60	0.84	0.60
Content Manager	0	0.69	0	0.30	0	0.47

Inverse Document Frequency (IDF) on the other hand is the logarithmic inverse of the document frequency for entire corpus such as out of 11244 user's profile, Mongo appeared in 521, calculated IDF is $log_{10}\left(\frac{11244}{521}\right) = 1.33$. Therefore, document frequency for skill Mongo, R, SQL, DBMS, HTML, and C is 1.33, 1.76, 2.2, 2.1, 2.09 and 2.3 respectively and similarly computed for all skills with respect to all 11244 user data. Table 8 shows computed TF-IDF score of user's skill corresponding to their job categories for user data.

Table 8. TF-IDF score of user skill according tojob category in user data

	Mongo	*R*	*SQL*	*DBMS*	*HTML*	*C*
Business Analyst	2.01	2.41	2.70	2.45	2.44	2.07
Human Resources	0.397	2.32	0	0.98	0	0
Software Developer	0	0.82	2.64	1.61	3.11	1.38
Web Analytics	1.02	0	2.2	1.26	1.75	1.38
Content Manager	0	1.21	0	0.63	0	1.08

6.2. Accuracy evaluation of proposed model:

In this section, we present the evaluation results on a manually tagged data set to validate the performance of recommendation system. We have taken a pre-tagged data set of about 1800 users which is16% of the total. Figure 3 represents frequency distribution of user designations in the validation dataset. Example – total "Human Resources" and "Business Analyst" in this dataset are 148 and 108 respectively.

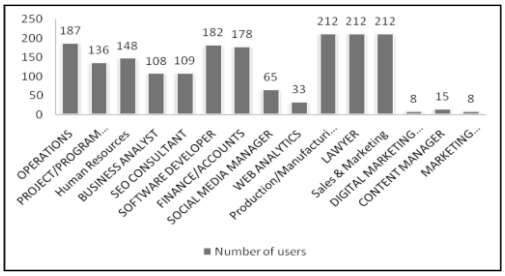


Fig 3. Frequency of users according to their designation

Further, we present the results of proposed topic modeling driven content based job recommendation system for every user designation. we tested the accuracy of our proposed approach using performance measures. The accuracy measurement metric has been used here to measure the performance of the system are - precision, recall and F-measure [15, 19]. Proposed topic modeling based content recommendation approach attains average 95% precise result. Mean recall and F1-measure on the taken 15% data set is 82% and 84 % respectively. Figure 4 shows the performance result.

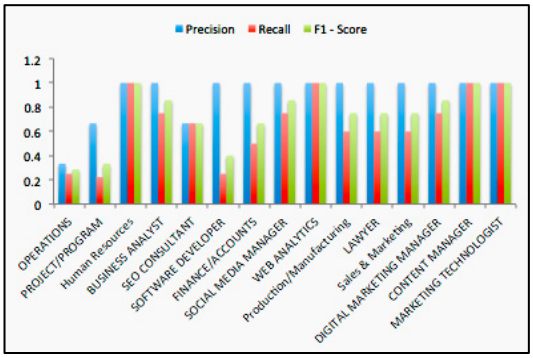


Fig 4. Performance/ accuracy measure of recommended job role corresponding to user designation

7. Conclusion

It is the need of current scenario to propose efficient recommendation algorithms with the help of machine learning and natural language processing algorithms. Although, to handle complex recommendation issues of varying application domains is also a challenging task. Therefore, we proposed latent semantic based recommendation approach to recommend relevant job postings to users based on their interests and skill set. Our study showed that latent feature based recommendation approach works well relative to traditional recommendation approach and including n-grams in feature set helped in increasing its accuracy. Proposed recommendation system is giving relevant predictions on completely new job postings. Further, our algorithm provides interpretable user profiles. and this could be useful in real-world recommender systems. For example, if a particular user would like recommendation according to specific skill set, system is able to answer according to chosen skill set. The results have shown the extremely adaptable evaluation score.

References

[1] David M. Blei, Andrew Y. Ng, Michael I.,Jordan, Latent Dirichlet Allocation, Journalof Machine Learning Research (2003) 993-1022
[2] Michael D. Ekstrand, John T. Riedl, Joseph A.Konstan, 2010, Collaborative FilteringRecommender Systems, Foundations and Trends inHuman–Computer Interaction Vol. 4, No. 2 (2010)81–175.
[3] R. Arun, V. Suresh, C. E. Veni Madhavan, M.N. Narasimha Murthy, 2010, On Finding the Natural Number of Topics with Latent Dirichlet Allocation: Some Observations, Springer BerlinHeidelberg
[4] Lu, Y., El Helou, S., & Gillet, D. (2013, May). A recommender system for job seeking and recruiting website. In Proceedings of the 22nd International Conference on World Wide Web (pp. 963-966). ACM.
[5] David M. Blei, Andrew Y. Ng, Michael I.,Jordan, 2003, Latent Dirichlet Allocation, Journalof Machine Learning Research 993-1022.
[6] Michael D. Ekstrand, John T. Riedl, Joseph A.Konstan, Collaborative FilteringRecommender Systems, Foundations and Trends inHuman–Computer Interaction Vol. 4, No. 2 (2010)81–175.
[7] Lu, Y., El Helou, S., & Gillet, D. (2013, May). A recommender system for job seeking and recruiting website. In Proceedings of the 22nd International Conference on World Wide Web (pp. 963-966). ACM.
[8] Sang Su Lee, Tagyoung Chung, Dennis McLeod, Dynamic Item Recommendation by TopicModeling for Social Networks.
[9] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. In Information Processing and Management, pages 513–523, 1988.
[10] Goeldi, Andreas. "Website network and advertisement analysis using analytic measurement of online social media content." U.S. Patent No. 7,974,983. 5 Jul. 2011.
[11] Cataldi, Mario, Luigi Di Caro, and Claudio Schifanella. "Emerging topic detection on twitter based on temporal and social terms evaluation." Proceedings of the Tenth International Workshop on Multimedia Data Mining. ACM, 2010.
[12] Cannas, L. M., Dessì, N., & Pes, B. (2013). Assessing similarity of feature selection techniques in high-dimensional domains. Pattern Recognition Letters, 34(12), 1446-1453.
[13] Naikal, N., Yang, A.Y., Sastry, S.S., 2011. Informative Feature Selection for Object Recognition via Sparse PCA, Proceedings of the 2011 International Conference on Computer Vision, 818-825
[14] Koudas, N., Marathe, A., & Srivastava, D. (2004, August). Flexible string matching against large databases in practice. In Proceedings of the Thirtieth international conference on Very large data bases-Volume 30 (pp. 1078-1086). VLDB Endowment.
[15] Huerta, J. M. Vector based Approaches to Semantic Similarity Measures. Advances in Natural Language Processing and Applications, 163.
[16] De Pessemier, T., Vanhecke, K., & Martens, L. (2016, September). A scalable, high-performance Algorithm for hybrid job recommendations. In Proceedings of the Recommender Systems Challenge (p. 5). ACM.
[17] ACM - Xing. RecSys Challenge 2016, 2016. Online available at https://recsys.xing.com/. Hoffman, M., Blei, D., Bach, F. On-line learning for latent Dirichlet allocation. In Neural Information Processing Systems (2010).
[18] Arora, A., Taneja, V., Parashar, S., & Mishra, A. (2016). Cross-domain based event recommendation using tensor factorization. Open Computer Science, 6(1).
[19] Behl, D., Handa, S., & Arora, A. (2014, February). A bug mining tool to identify and analyze security bugs using naive bayes and tf-idf. In Optimization, Reliabilty, and Information Technology (ICROIT), 2014 International Conference on (pp. 294-299). IEEE.