A COMPARATIVE STUDY ON CONTENT-BASED PAPER-TO-PAPER RECOMMENDATION APPROACHES IN SCIENTIFIC LITERATURE

Bahareh Kazemi
Department of Computer Science
Ryerson University
350 Victoria St
Toronto, ON M5B 2K3
bahareh.kazemi@ryerson.ca

Abdolreza Abhari
Department of Computer Science
Ryerson University
350 Victoria St
Toronto, ON M5B 2K3
aabhari@scs.ryerson.ca

ABSTRACT

This paper deals with analysis and comparison of two well-known content-based recommendation approaches for scientific papers in biomedical domain. Given a rich set of abstracts for thousands of articles from PUBMED, a series of efficient pre-processing techniques are proposed. Then, for the first approach, a Term-frequency Inverse-document-frequency (TF-IDF) algorithm is formulated to make recommendations for the paper-set. Alternatively, we also use word-embedding to represent papers' abstract text and employ the extracted representation for the recommendation construction. Experimental results will evaluate and compare the efficiency and suitability of any of the proposed frameworks in building a universal paper-to-paper recommendation engine.

Keywords: content-based recommendation, TF-IDF, data preparation, word-embedding

1 INTRODUCTION

Nowadays, online scientific literature is one of the most important resources for all the researcher around the world. Considering large volume of papers published in any specific field, having a scientific network of these papers that also defines some meta-relations among each other can provide useful data to aim researchers finding their target information. On the other hand, it is observed that the number of available references matching a certain requirement is growing drastically and that brings another challenge for the researchers on how to find the most relevant papers to their desired concept in the least amount of time. To fulfill both requirements, different families of recommender systems have been designed in recent years.

Collaborative-filtering based recommender systems have been widely used for design and implementation of recommendation engines in different fields (Cacheda et al. 2011, Lee et al. 2012). The original idea is based on how to recommend a product to a certain user based on the same item purchased or rated by similar users. The same idea has been brought to the scientific literature network by mapping the [user-product] tuple to the relevant entities (e.g. paper to field of study,…). In (Gipp. 2010), authors have leveraged scientific citation network (shows which paper cites or is cited to or by other papers) of papers as initial paper-to-paper relations and, then, find top recommended papers based on the papers cited by the most similar papers to the original one. A modified version of the above algorithm is also proposed in (Kukuktunk et al. 2012) by updating the probability of visiting a new paper recursively. The main idea is to adapt the well-known page-rank (Page. 2001) algorithm and build a direction-aware based technique on-top-of the citation network. Collaborative-filtering has been also developed in (Basu et al. 2001) for concept-to-concept recommendation. The main idea is to form a concept-to-concept network by co-

SpringSim-CNS 2017, April 23-26, Virginia Beach, VA, USA

pairing all concepts mentioned in any of two papers appearing in the citation network. To clarify more, given that each paper covers a number of concepts, citing a paper by another paper means they are conceptually related and that's where a concept-to-concept relation can be established. Collaborative-filtering is then applied to the new formed network to recommend most similar concepts.

While collaborative-filtering methods work based on the paper citations, they may suffer from sparsity in papers co-occurrences. Another family of recommendation systems then considers co-cited papers within a single paper. Co-citations are also scored based on the proximity of their occurrence. In (Gipp. 2010), a co-citation proximity based technique has been developed for paper-to-paper recommendation. It is shown that by proper scoring the co-citations, the algorithm provides more relevant results compared to the traditional collaborative-filtering, which solely utilizes the citation-network.

While both above approaches only consider the co-cited papers in making the recommendations, they do not take the papers' content into account. The main goal in content-based information is then to represent paper's content as a vector and calculate similarity of papers to enhance the recommendation. A review of state-of-art content-based algorithms has been made in (Beel et al. 2016). It is mentioned that half of literature is now focusing on content-aware recommendation. In (Philip et al. 2014), authors use Term-frequency Inverse-document-frequency (TF-IDF) to represent papers and user queries. After calculating similarity of query/paper pair, the algorithm recommends most relevant papers to each user query. A family of TF-IDF based recommenders has been also reviewed in (Junior. 2004). Authors have also combined TF-IDF based method with collaborative-filtering and came up with a new hybrid version of recommendation system.

This paper mainly focuses on implementing and comparison of two well-known content-based techniques for paper-to-paper recommendation problem. After collecting abstract information from papers indexed by PUBMED (PUBMED), two techniques are used to represent papers' text (abstract) for recommendation purposes. Beside the traditional TF-IDF, we use word embedding (Mikolov et al. 2013) to construct a paper embedding that results in a much more compressed word representation too. The paper embedding vector is also generated as summation of individual words' embedding within the paper. Both above sets of vectors are then separately used to make top recommendations for each target paper and efficiency of recommended results are ultimately compared.

The rest of this paper is organized as follows. Section 2 discusses content-based recommendation. The basic idea behind the algorithm is explained and both TF-IDF and embedding-based representations are described. Section 3 is then dedicated to the experimental studies. First, different steps of data preparation and cleaning are discussed. Then, two versions of the algorithms, one based on TF-IDF and the other using embedding, are applied to the PUBMED abstract set. Some metrics are also presented to compare the performances. Concluding remarks are finally given in section 4.

2 CONTENT-BASED RECOMMENDATION

2.1 Problem Statement

Given a set of papers as $P=\{p_0 , p_1 , p_{2,\dots}p_n\}$ and a set of paper's abstract as $T=\{t_0,t_1,t_2,\dots,t_n\}$ where t_m represent the abstract text of paper p_m, the goal is to provide top most relevant papers for any paper p_m. To formulate this, it is needed to represent and map T to a set of vectors such as $V=\{v_0 , v_1, , v_2 ,\dots, v_n\}$, where v_m corresponds to paper p_m 's abstract text(t_m), and from there we can apply desired algorithms and calculate the similarity.

In the following, first we dig into vector representation with the use of TF-IDF and then, vector representation based on word-embedding (a Neural-network based approach) will be reviewed.

2.2 A TF-IDF Based vector representation

Each paper can be mathematically represented with use of TF-IDF vector. To achieve this, we first need to create a dictionary of all words from all the paper's raw text (abstract). In this paper, words are achieved as the result of applying a white-space tokenizer to the paper's text. Assuming V as the size of the dictionary, TF-IDF vector can be calculated by following these steps below:

a) For every single paper, TF is defined as a $V \times 1$ vector of integers. The v-th entry in the vector specifies how many times the dictionary's v-th word appears in that paper. Expectedly, this TF term is unique for each paper.
b) On the other hand, assuming N as the number of all papers in the pool, IDF is a $V \times 1$ vector with v-th entry defined as $[idf]_v = \log\frac{N}{n_v}$ with n_v being the number of papers that contain the v-th word.
c) TF-IDF: finally, TF-IDF is defined as a $V \times 1$ vector with the v-th entry being calculated as $[tf]_v \times [idf]_v$.

Getting each paper and its associated text (p_n, t_n), where t_n is the abstract representation of paper p_n, above procedures can be proceeded to generate a vector τ_n as the TF-IDF representation of the n-th paper.

2.3 Word-embedding based vector representation

The TF-IDF based representation suffers from a number of issues. First, TF-IDF vectors do not consider semantic similarity of documents. Indeed, only documents with a high volume of shared words (exact matches of words) are taken similar in a TF-IDF context. Therefore, articles with different sets of words, but very correlated semantic meanings, may get a very low similarity score based on TF-IDF, which causes loss of valuable recommendation results.

In addition, given the local nature of TF-IDF representations (Bengio et al. 2013),which normally results in storing big size dictionaries, their usage usually leads to a huge amount of disk usage. As the size of the TF-IDF vector varies by the number of words in the dictionary, the similarity calculation also requires a vast amount of time given a long representation of TF-IDF vector. Other notorious issues in using local representations like TF-IDF can be found in (Bengio et al. 2013).

Recently, with emerge of artificial neural networks, compact word representations using vector embedding has become popular. The well-known word2vec framework (Mikolov, et al. 2013) finds a unique embedding vector for each word in the dictionary, where each word is generated by white-space tokenization of the raw-text and the dictionary is constructed as a collection of generated unique words. Two popular structures of word2vec are provided in Figure 1. The idea in both approaches is that neighbor words within a document are semantically related. Therefore, the structures in Figure1 are trained using a collection of word neighbors where the central word is used as the ground truth (the network is designed to predict the central word given its neighbors). The trained network can be then used to encode each given word in the dictionary and output a word embedding, which is much more compact than a TF-IDF representation.

For the purpose of content-based recommendation, we represent each paper by an embedding vector where the vector is calculated as summation of the embedding of each individual word appearing in the paper. Algorithm1 summarizes different steps for generating the paper embedding. While embedding vectors are calculated, they can be used for calculating similarities between each two papers.

2.4 Algorithms for content-based recommendation

Now that each paper is represented by a vector, the content-based recommendation algorithm can be constructed as shown in the Algorithm 2. The given algorithm is based on the TF-IDF vector representation. However, the algorithm can be easily adopted for the embedding-vector representation

where the similarity score is now found based on the vectors calculated using embedding (Table 1 presents the algorithm using word-embedding). Note that, unlike TF-IDF based approach, we do not restrict the set of candidate papers for similarity calculation in the embedding-based framework. Unlike the TF-IDF representations, embedded vectors are more compact in size and similarity calculation using them will be more efficient. Therefore, for each given target paper, the similarity is calculated for any other paper in the set when embedded vectors are provided.

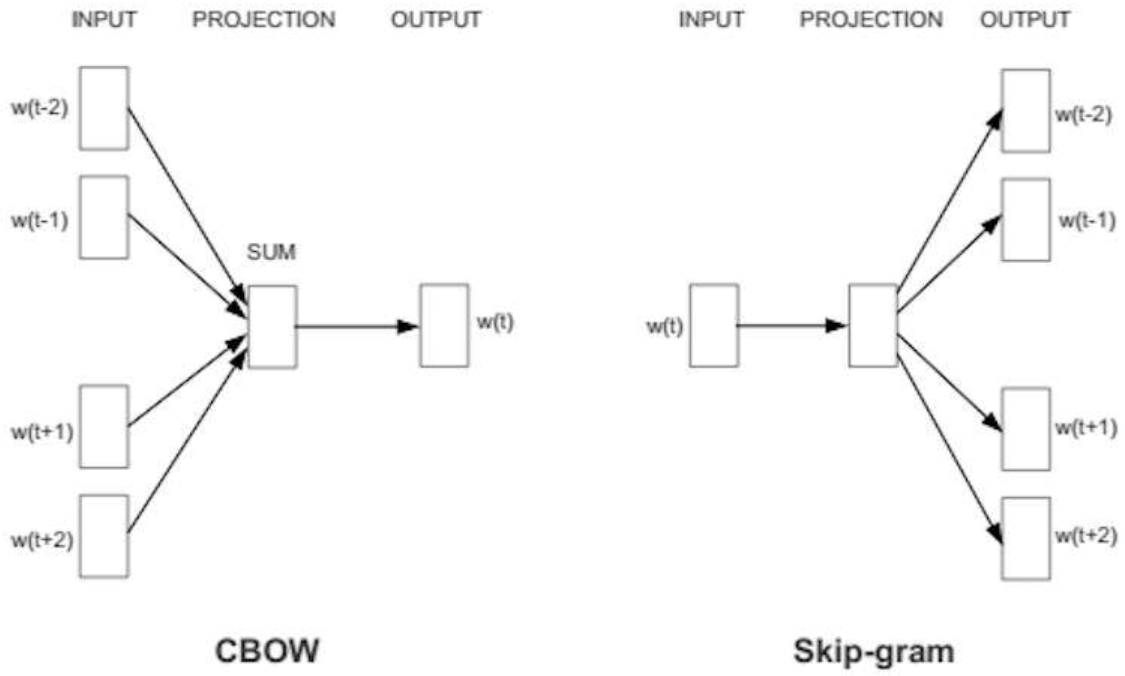

Figure1: Two different structures for modeling and extracting word embedding (Mikolov et al. 2013). The left-hand structure (CBOW) takes neighbors of each word as the input to a neural-net and predicts the word in the output. The right-hand model (Skip-gram) gets each word and intends to generate its neighbors in the output layer.

Algorithm1: paper-embedding extraction

1. Inputs: set of all text (abstract) of papers
2. Outputs: vector embedding for each paper
3. Text tokenization: a white-space tokenizer is applied to texts to extract all the words
4. Dictionary construction: create a dictionary out of all input texts containing all unique words after tokenization
5. Training data generation: search in each paper and extract set of neighbor words as $(w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2})$ with w_i being the ground-truth, where i denotes the spatial index of the word within the text
6. Train a CBOW model using all generated records and extract embedding for each word in dictionary ($\boldsymbol{v}_n$; the embedding representation of n-th word in the dictionary)
7. For each paper, calculate embedding representation as a summation of all individual words' embedding inside the paper

8. Similarity calculation:
    a. For each target paper p_m : calculate $sim(\boldsymbol{v}_{p_m}, \boldsymbol{v}_{p_s})$ as the cosine-similarity of paper's embedding representations, where p_s is all remaining papers in the network
    b. Rank all similarity scores and pick top-N papers $(p_{s_1}, \dots, p_{s_N})$ from the ranked-list

Algorithm2: TF-IDF based algorithm for paper-to-paper recommendation

1. Inputs: a target paper (p_n) and set of all papers' abstract
2. Outputs: set of top-N recommended paper for the target paper (N is a parameter)
3. Text tokenization: a white-space tokenizer is applied to abstracts to extract all the words
4. Dictionary construction: create a dictionary out of all input abstracts such that it contains all unique word after tokenization
5. IDF calculation: calculate IDF for each word in the dictionary
6. TF-IDF calculation
    6.1. Choose S as the set of all papers assigned to the target paper (initially, a null set)
    6.2. For each word in target paper p_n:
        6.2.1. If IDF(word)>α: $S = S \cup${set of all papers containing that word}
    6.3. For each paper in S: calculate TF-IDF
7. Similarity calculation:
    7.1. For each paper in S: calculate $sim(p_n, p_s)$ as the cosine-similarity of TF-IDF representations
    7.2. Rank all similarity scores and pick top-N papers $(p_{s_1}, \dots, p_{s_N})$ from the ranked-list

3 EXPERIMENTAL RESULTS

3.1 Data Cleaning and Preparation

In order to operate this experiment we used the data provided by PUBMED (PUBMED). While PUBMED has released abstract text for up to 25 million papers in the areas of medicine and biology, for the purpose of this paper, a subset of hundred-thousand papers is randomly selected out of all papers that are published between 2010 and 2015 in PUBMED website. Then, the following processes are applied to the raw abstract text:

1. Each abstract is tokenized using a simple white-space tokenizer and all tokens are then retrieved.
2. All punctuations are removed and any word containing numerical values (pure numerical or alpha-numerical) is deleted.
3. Each cleaned word is also lemmatized and stemmed using standard word-net lemmatizer and English stemmers.

Applying above steps to the raw-abstract, a dictionary of 80000 words is constructed, which will be used for TF-IDF and embedding calculations.

3.2 Content-based recommendation using TF-IDF

At this stage, for the purpose of result evaluation, ten papers are randomly chosen out of the paper-pool as the target papers. These papers will be used to generate recommendation list, which will be used to evaluate the efficiency of any of proposed methods. For each paper, ten papers are displayed as the

recommendations. As briefly mentioned before, for each target paper, the algorithm creates a vector of unique words out of the abstract and filter those with IDF less than an arbitrary threshold. Note that IDF threshold is used to delete common words and reduce the dimension of TF-IDF vector. Here, we picked the threshold to be 6 and the algorithm will then search only for the papers that contain words whose IDF meets the threshold. Note that there is a trade-off between the choice of the threshold and the computational complexity as the lower the threshold, the larger the paper-pool. This approach would help to deal with a smaller size of paper pool and produce the result quicker. The size of the pool that the algorithm is experiencing is by average one thousand (varies from 300 to 1500), which is reasonably a good number considering there are one hundred thousand papers in the network and, on average, it takes roughly fifteen minutes to produce the sorted recommended papers.

***For instance, here the results are shown during a random round of testing. Below shows the id and the abstract text of the target paper that was selected :

57749 Epidemiology of head and neck cancer. This article discusses risk factors, incidence trends, and prognostic considerations for head and neck cancer (HNC). The primary causes of HNC are tobacco and alcohol use, and human papillomavirus (HPV). Tobacco-related HNC incidence rates are decreasing in countries where tobacco use has declined. HPV-HNC, which occurs primarily in the oropharynx and is associated with sexual behaviors, has been increasing over the past several decades, among white men in particular. The prognosis for HNC overall has improved slightly since the 1990s, and is influenced by site, stage, and HPV status. Prognosis for HPV-HNC is significantly better than for HPV-negative disease.

And for this target paper, the id of top ranked papers in term of relevancy and their corresponding similarity rates are as below:

Table 1: Similarity scores.

Rank	Paper id	Similarity score with target paper
1	10768	0.548438963796
2	25034	0.521074854832
3	93457	0.485960717857
4	79513	0.375827831579
5	78645	0.34136517326
6	66967	0.332363622782
7	15035	0.299527681887
8	73036	0.236020770353
9	24656	0.230798191207
10	1133	0.230775326915

The abstract text for the top-recommended paper can be presented as follows:

10768 S-Nitrosylation of mitogen activated protein kinase phosphatase-1 suppresses radiation-induced apoptosis. Radiotherapy is a key modality for head and neck cancer (HNC) treatment. Mitogen activated protein kinase phosphatase-1 (MKP-1) protein levels are elevated in various tumors and are negatively correlated with efficacy of chemo- or radio-therapy. However, the mechanisms underlying the moderate radiosensitivity of HNC and the increased MKP-1 protein levels are still dismal. Here we show that S-nitrosylation of MKP-1 on Cysteine 258 enhances MKP-1 protein stability, phosphatase activity, and MKP-1-mediated anti-apoptotic effect on HNC radiotherapy. Co-culturing MKP-1 transfected HNC cell lines with activated macrophages for mimicking the microenvironment of the irradiated cancer cells further confirms that S-nitrosylation-mediated increase of MKP-1 activity correlates with decrease of HNC radiosensitivity. Therefore, S-nitrosylation of MKP-1 presents a novel mechanism underlying the enhanced MKP-1 expression levels and MKP-1-mediated radio-resistance in head and neck cancer.

As it is clear from the results, the first significant factor is the low relevancy rates for the top recommended papers (with having only 0.55 similarity score for the top recommended one). This can be due to the nature of the TF-IDF method that needs reasonably good size text, for instance a full text of a paper, to be able to provide better recommendation based on similarity. Moreover from the above shown recommended abstract, it can be concluded that the recommendations are not exactly conceptually related and they are selected since they have shared a number of keywords in their abstract with target paper.

3.3 Content based recommendation using embedding

The previous section summarized recommendation results using TF-IDF based vector representation. In this part, we represent each paper by a new vector extracted from individual word embedding mentioned in the paper. The first step here is to represent each word in the dictionary by a vector called embedding. To do this, a CBOW model as shown in figure 1 is trained using training records like $(w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2})$. Here, the window of five words is generated from the paper's abstract where words are assumed to be next to each other within a sentence or paragraph. It is worth to mention that, in the constructed CBOW model, $(w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2})$ represents the input training-set and w_i denotes the target output. Processing all papers in the pool-set, 770000 training records are generated. The CBOW model is then trained and each word's embedding is calculated as the output of the input layer of the CBOW model. It is then expected that semantically close words achieve more similar word embedding which results in higher cosine similarity. As an example, consider the word **hospital** and top-10 close words based on the cosine-similarity of vector embedding. The list is given as:

{care,cost,institute,department,photocatalysi,admiss,stay,supervisori,pharmacy}

It is observed that the trained CBOW model has been able to connect words with close meanings properly. The above test can be similarly conducted by other words in the dictionary.
In the next step, the vector embedding for each document is calculated as a summation of each individual word's embedding in the text. Generating all papers' embedding, the algorithm in Table 2 is implemented to generate top-N recommended papers for each target-paper. We use the same set of target papers used in the last section to generate recommendation results using papers' embedding. The list of top-10 recommended papers for paper-id=57749 (the exact the same id as used for TF-IDF case) can be observed in below:

Table 2: Similarity scores.

Rank	Paper id	Similarity score with target paper
1	87181	0.90554864404327018
2	62869	0.89667078996558947
3	59282	0.8963773679685213
4	23179	0.89477514700716965
5	79458	0.89420929374297575
6	90844	0.8923484056428060
7	87839	0.89177949310457949
8	68193	0.89158001108560014
9	66131	0.88890781814655662
10	11448	0.88877877080517143

The top-recommended paper abstract based on the vector embedding can be also given as follows:

Health-related behaviours in the EpiPorto study: cancer survivors versus participants with no cancer history. Cancer survivors are at an increased risk of a second primary cancer, partly due to unhealthy behaviours. In a cohort of adults (recruitment: 1999-2003; follow-up - linkage with population-based cancer registry: up to 2009) we compared the baseline exposure to smoking, alcohol and dietary intake and physical activity between: cancer survivors (CS) - cancer diagnosis before baseline (n=53); no cancer (NC) participants - without cancer diagnosis at baseline or during follow-up (n=2261); latent cancer (LC) participants - without cancer diagnosis at baseline but diagnosed during follow-up (n=139). Age-, sex- and education-adjusted prevalences and means were computed, as applicable. The prevalence of current smoking was nearly 20% among CS and NC (approximately four cigarettes per day) and 30% in LC (seven cigarettes per day). LC had the highest average alcohol intake (25.5 g/day) and NC the lowest (17.0 g/day). The proportion of participants reporting sports practice was higher for CS (50%) than for NC or LC (approximately 33%). CS and NC had higher fruit/vegetable consumption than LC (4.2 and 4.4 vs. 3.8 servings per day). In a composite index on health behaviours (including smoking, physical activity and alcohol and fruit/vegetable intake) the highest and lowest scores were 1.74 for NC and 1.52 for LC respectively, whereas CS scored 1.63. The exposure to each risk factor appeared comparable in CS and NC, whereas LC tended to have unhealthier behaviours. This may be partially explained by the acquisition of healthier habits by CS after diagnosis, but there still remains scope for improvement, as revealed by the low scores observed for the joint exposure to the main risk factors

3.4 Numerical comparison of embedding and TF-IDF approaches

In order to compare the efficiency of any of the proposed algorithms, the similarity scores of each pair of papers in the recommendation list are calculated using Microsoft Academy Knowledge Graph (https://www.microsoft.com/en-us/research/project/microsoft-academic-graph). The designed API by Microsoft takes abstracts of each two papers in the recommendation list and calculates a similarity score based on various semantics and word matchings in the abstracts. The scores are then calculated for both recommended sets and a set of scores is generated for each recommended paper and the target one. The average, maximum and minimum of the generated set of scores for each recommendation set are then reported as the final performance metrics. The results for any of approaches are shown in Table 3. As observed, embedding recommendations provide at least 15% higher accuracy compared to TF-IDF for all metrics. Indeed, embedding-based approach has provided more semantically-relevant papers to the target paper, which leads to more accurate recommendations.

Table 3: Similarity scores for TF-IDF and embedding-based recommendations.

Metric	Average	Minimum	Maximum
TF-IDF	.5945	.48	.7417
Embedding	.7168	.55	.92

Based on the embedding-based results, it can be observed that:

- The lists of top-10 recommended results do not intersect well for embedding and TF-IDF based methods. This is due to the nature of approaches any of these two techniques leverage to generate top-N recommendations. While TF-IDF approach works based on the similarity of papers' abstract according to the exact mention of terms in the pair, embedding approach takes the semantic relatedness of papers into consideration as well. That is, papers with different lexical mentions may still be very related as set of words are closely related in terms of semantic or meaning (e.g. denote, correspond and mean are all semantically close). That's why embedding approach provides a broader recommendation set beyond just similar lexical content.
- As word embedding provides a much more compact representation of words and papers, it achieves a more scalable solution compared to the TF-IDF based technique. No matter how big the exact size of abstract is, each abstract is represented by a vector with the size of word embedding vector. In this case, for hundred of thousand of papers, we chose the size of embedding to be 100.There should be a reasonable relation between the number of all papers that are supposed to be shown by an embedding vector and the size of the vector. For instance, clearly, a vector of size 5 is not sufficient to represent hundred of thousand of papers. For running the experiment, it takes only 2 minutes to recommend top-10 papers to each target paper while TF-IDF based requires more than 10 minutes to provide a sensible recommendation set.
- As a comparison base-line, similarity scores from Microsoft Academy Graph also shows more relevant recommendations by embedding-based algorithm as the similarity scores provided by Microsoft API shows higher values for the embedding approach compared to the TF-IDF based technique.

4 CONCLUSIONS

A content-based recommender system and two of its widely used sub-methods have been reviewed in this paper for the purpose of paper-to-paper recommendation problem. By reviewing the presented results, it is observed that the embedding method produces more accurate (more relevant papers) recommendation results in comparison to TF-IDF based approach. In addition, given the compact nature of embedding vectors, the embedding-based framework provides a more scalable platform than the TF-IDF based

paradigm. By using an API we found embedded wording in average generates 22% more accuracy than TF-IDF method. We verified the outperformance of embedding wording over TF-IDF by manual comparison of the top ranked recommended papers in both methods. It is also worth to note that in order to reach more reliable and accurate comparison between two above mentioned approaches, there can be a group of scientists who work in the related fields hired and simultaneously provide them with the target and recommended papers and ask them to rate the relevancy of the results for each of the approaches. This part is considered as one of future works to provide a solid comparison of traditional TF-IDF based methodology and an approach based on the embedding.

REFERENCES

Cacheda F., V. Carneiro, D. Fernandez, and V. Formoso. 2011. "Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender systems". ACM Transactions on the Web, Vol. 5, Issue 1.

Lee J., M. Sun, G. Lebanon. 2012. https://arxiv.org/abs/1205.3193**.**

Gipp B., "Measuring Document Relatedness by Citation Proximity Analysis and Citation Order Analysis", in Research and Advanced Technology for Digital Libraries: Proceedings of the 14th European Conference on Digital Libraries (ECDL'10), 2010.

Kukuktunk O., Erik Saule1, Kamer Kaya1, Ümit V. Çatalyürek1. 2012. "Recommendation on Academic Networks using Direction Aware Citation Analysis", 6th Int'l Workshop on Ranking in Databases (DBRank) in conjunction with VLDB'12.

Page L.. 2004. "Method for scoring documents in a linked database, US 6,799,176 B1.

Basu C., H.Hirsh, W.W.Cohen. 2001. "Technical Paper Recommendation: A Study in Combining Multiple Information Sources", Journal of Artificial Intelligence Research 1 231-252.

Beel J., B. Gipp, S. Langer, and C. Breitinger. 2016. "Research-paper recommender systems: a literature survey". International Journal on Digital Libraries. Vo. 17, Issue. 4.

Philip S., P. B. Schola, and A. O. John. 2014. "Application of content-based approach in research paper recommendation system for a digital library". International Journal of Advanced Computer Science and Applications. Vol. 5, No. 10.

Junior R. D. T.. 2004. "Collaborative filtering and content-based filtering to recommend research papers". MSc Thesis. Universidade Federal Do Rio Grande.

Mikolov T., I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013. "Distributed Representations of Words and Phrases and their Compositionality". Advances in neural information processing systems. 3111-3119.

Bengio Y., A. Courville, and P. Vincent. 2013. "Representation Learning: A Review and New Perspectives". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 35, Issue. 8, pp. 1798-1828.

Pubmed. https://www.ncbi.nlm.nih.gov/pubmed/.

AUTHOR BIOGRAPHIES

BAHAREH KAZEMI is a affiliated with the Department of Computer Science in Ryerson University.

ABDOLREZA ABHARI is a affiliated with the Department of Computer Science in Ryerson University.