Reranking-based Recommender System with Deep Learning

Ahmed Saleh [1,2], Florian Mai [1,2], Chifumi Nishioka [1,2], Ansgar Scherp [1,2]

Abstract: An enormous volume of scientific content is published every year. The amount exceeds by far what a scientist can read in her entire life. In order to address this problem, we have developed and empirically evaluated a recommender system for scientific papers based on Twitter postings. In this paper, we improve on the previous work by a reranking approach using Deep Learning. Thus, after a list of top-k recommendations is computed, we rerank the results by employing a neural network to improve the results of the existing recommender system. We present the design of the deep reranking approach and a preliminary evaluation. Our results show that in most cases, the recommendations can be improved using our Deep Learning reranking approach.

Keywords: recommender systems; deep learning; semantic profiling

1 Introduction

The increasing volume of published scientific papers increases the need for recommender systems. In our earlier work, we have studied the possibility of building a title-based recommender system and compared it to a recommender system which analyses the full-text for making scientific paper recommendations [NS16]. To this end, we have introduced a novel hierarchy-based scoring model, called HCF-IDF (Hierarchical Concept Frequency - Inverse Document Frequency), to compute the list of top-k recommendations. HCF-IDF extends the classical information retrieval model TF-IDF (Term Frequency – Inverse Document Frequency) by the use of semantic concepts (C) that are organized in a poly-hierarchical knowledge graph (H). In an online experiment with $n = 123$ Twitter users in the area of economics, we could show that our recommender system and novel scoring function can provide competitive scientific paper recommendations based only on the titles of the papers versus analysing the entire fulltext. A variant of HCF-IDF is CF-IDF (Concept Frequency - Inverse Document Frequency), which results from leaving out the hierarchical relations in the knowledge graph.

In this paper, we extend on our previous work by investigating to further improve the recommendation quality by reranking the results using a Deep Learning technique. The reranking is based on Paragraph Vectors [LM14], a popular technique to compute a fixed-length vector representation of a variable-length input text. In contrast to classical

[1] ZBW – Leibniz Information Centre for Economics, Kiel and Hamburg, Germany, {stu200671, stu96542, chni, asc}@informatik.uni-kiel.de
[2] Knowledge Discovery, Department of Computer Science, Kiel University, Kiel, Germany

Bag-of-Words representations, Paragraph Vectors take the sequence of word occurrences into account and can represent word similarities. As dataset, we employ the evaluation results of our online user study carried out earlier [NS16]. It contains for each user and recommendation method like HCF-IDF and CF-IDF a precomputed list of top-k recommended scientific publications and their relevance. On these recommended lists, we apply Paragraph Vectors to rerank the results. We measure the improvement given the gold standard from the online study using the information retrieval metric nDCG (normalized Discounted Cumulative Gain).

2 Literature Review

Capturing the user interests from a set of text updates (e.g., tweets, social media status) has been studied in different works. For example, Chen et al. presented a recommender system that could recommend URLs to the users based on their own Twitter profiles and social voting [Ch10]. Goosen et al. proposed a recommender system for news articles based on a novel CF-IDF profiling method. The authors performed an experiment where 100 news articles have been displayed to 19 users who then had to indicate whether the article was relevant or not [Go11]. The results show that CF-IDF outperformed the popular TF-IDF model. Inspired by this work, we have developed HCF-IDF, a hierarchical variant of CF-IDF [NS16]. We have built a recommender system based on twelve user profiling strategies, each of which generated five recommendations to 123 participants in an online experiment. Half of the strategies were applied on full texts and the other on titles of scientific publications. The participants assessed whether the recommendations were of interest to them or not. The results showed that the best recommendations were achieved by CF-IDF and HCF-IDF on full-texts, with a sliding window as a decay function. However, the novel HCF-IDF profiling method achieves similar results just by using the titles of the publications. Following the successful application of HCF-IDF, we have investigated the possibility of building a deep learning model that captures syntactic and semantic word relationships in the users' social media profiles in order to provide better recommendations. To this end, we investigated recent word embedding techniques. Mikolov et al. introduced the skip-gram model, an efficient method for learning high quality vector representations of words from large amounts of unstructured text data [Mi13]. The principle idea of skip-gram models is to generate the word vectors. The approach demonstrated great performance for explicitly encoding different linguistic regularities and patterns. The results inspired Le and Mikolov to develop an unsupervised learning technique, called Paragraph Vectors, that takes advantage of the skip-gram model's words representation to learn fixed-length feature representations from texts of different lengths (e.g. documents or users' social media profiles) [LM14].

3 Method Description

We start by discussing methods for profiling documents and computing user profiles. Subsequently, we describe how to use the results of these methods as an input for a cosine similarity function in order to provide an initial ranking of top-k relevant documents. Finally, we present an approach for reranking the top k relevant documents using the deep learning Paragraph Vectors model.

3.1 Profiling methods

We present the two methods for profiling, which provided the best recommendations in our previous study, namely HCF-IDF and CF-IDF [NS16]. **CF-IDF** is an extension of TF-IDF that counts concepts instead of terms. CF represents the frequency of occurrence of a concept, while the IDF factor of a term is inversely proportional to the number of documents in which the concept appears. This means, the lower the concept appears in the corpus, the higher the IDF factor and vice versa. CF-IDF is the product of these quantities and is calculated using Equation 1:

$$score_{cf-idf}(c,d) = freq(c,d) \cdot \log \frac{|D|}{|d \in D : c \in d|} \tag{1}$$

$|D|$ denotes the number of documents in our dataset and $|d \in D : c \in d|$ are the number of documents d in D containing c. **HCF-IDF** is an extension of CF-IDF that takes into account the hierarchical structure of concepts by applying a spreading activation function. Figure 1 shows an example where a users' social media profile includes the concept "loss offset", a concept in a thesaurus for economics. Because of the hierarchical structure of the thesaurus of economics, HCF-IDF will also activate the parent concepts "corporate tax management" and "business economics". The algorithm will automatically give less weight to the general concepts, like "business economics", by using the bell log spreading activation which has been presented by Kapanipathi et al. [Ka14]. HCF-IDF is a product of the following two quantities and is computed using Equation 2:

$$score_{hcf-idf}(c,i) = BL(c,i) \cdot \log \frac{|I_u| + |I_r|}{|i \in I_u \bigcup I_r : c \in i|} \tag{2}$$

$|I_u|$ denotes the user's social media stream, $|I_r|$ is the number of random social media items, and $BL(c,i)$ is the bell log spreading activation function.

3.2 Ranking based on similarity scores

In order to calculate the similarity between the social media profile of the user P_u and the documents profiles P_d, which have been generated using the CF-IDF and HCF-IDF methods, we employ the cosine similarity using Equation 3:

$$\sigma(P_u, P_d) = \frac{P_u \cdot P_d}{||P_u|| \cdot ||P_d||} \tag{3}$$

Based on the similarity scores $\sigma(P_u, P_d)$, the top five documents are chosen and passed to a pre-trained Paragraph Vectors models for reranking the documents based on their semantics.

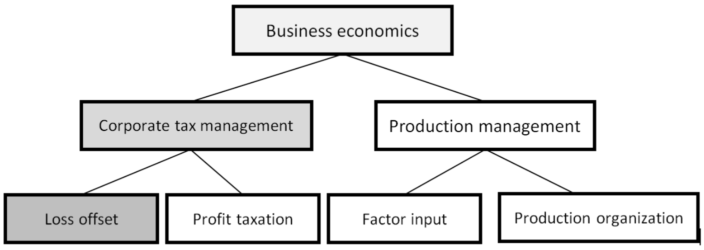

Fig. 1: Example of applying HCF-IDF and spreading activation for giving weights in a thesaurus for economics.

3.3 Reranking with Paragraph Vectors

The approach aims to map a text of arbitrary length ("paragraph") into a vector space in such a way that similar texts are located close to each other. In our experiment, we employ the Distributed Memory Model of Paragraph Vectors (PV-DM) [LM14]. The underlying assumption is highly influenced by the skip-gram word embedding models [Mi13], where one word vector is mapped into a vector space, in which semantically similar words have similar vector representations. To this end, it performs the task of predicting the center word of a sequence given its context: $\log P(w_t | w_{t-k} \ldots w_{t-1} w_{t+1} \ldots w_{t+k})$. The Paragraph Vectors model extends this model by optimizing a representation that encodes the probability of a sequence of words in a paragraph. Formally, we maximize the sum of all average log probability $\log P(w_t | w_{t-k}, \ldots, w_{t+k}, r)$ over all center words w_t occurring in paragraph r. In order to perform the prediction task, the softmax activation function (Equation 4) is used:

$$P(w_t | w_{t-k}, \ldots, w_{t+k}, r) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}} \tag{4}$$

where y_i is the log-probability for output word i, computed as in Equation 5:

$$y = b + U \cdot h(w_{t-k}, \ldots, w_{t+k}, r; \mathbf{W}, \mathbf{D}) \tag{5}$$

where U, b are the softmax parameters and h is a function that concatenates the word and paragraph embeddings extracted from **W** and **D** [LM14]. In order to train our Paragraph Vectors model, we used ZBW's economics dataset and Wikipedia dataset. ZBW's economics dataset consists of around 288,000 open access English publications while the Wikipedia dataset consists of more than 3,750,000 English articles. We trained a Paragraph Vectors model for each of the documents by moving a sliding window of $size = 5$ for the economics dataset and $size = 8$ for the Wikipedia dataset, over each sentence of the text. Contexts over multiple sentences are disregarded. We set the dimensionality of the feature vectors for the economics model to 100 and 1000 for the Wikipedia model. In order to speed up the training process, we used hierarchical softmax as suggested in [LM14]. Finally, the trained

Paragraph Vectors models are utilized to compute the vector representation of a document and the user's social media profile (concatenation of all of the user's tweets separated by a blank), and compute the cosine similarity between both of them. Two Paragraph Vectors will be similar (in terms of cosine similarity), if the encoded texts are similar. This is utilized to improve the recommendations based on the recommendations from the CF-IDF and HCF-IDF profiling methods by reordering the top five recommended documents.

4 Experimental Evaluation

In this section, we will present more information about our evaluation procedures in Section 4.2. The datasets which have been used are described in Section 4.1 and the results in correspondence with the nDCG evaluation metric are shown in Section 4.3.

4.1 Datasets

Economics thesaurus (STW) The economics thesaurus[1] contains a vocabulary of more than 6,000 economic subjects. This thesaurus is developed and maintained by an editorial board of domain experts at ZBW – Leibniz Information Centre for Economics.

EconBizRecSys evaluation dataset[2] contains the evaluation results from a previously built title and fulltext-based recommender system experiment with assessments of n = 123 participants after analysing their Twitter profiles [NS16]. The evaluation results have been generated from twelve different strategies, each of which contains five scientific publication recommendations and its corresponding user assessment. The recommended scientific publications are a subset of the ZBW Economics dataset with 288,000 English publications.

4.2 Procedure

In order to evaluate the performance of our recommender system, we have used the two datasets described above. The first one, STW, has been used for building the document profiles and the users' profiles. The second dataset has been used to re-evaluate the recommendation results of the HCF-IDF and CF-IDF profiling methods and evaluate the reranking recommendation results of the Paragraph Vectors model comparable to the users' assessment as a gold standard.

In order to evaluate the usefulness, often called 'gain', of the recommended document based on its position in the recommendations list, we apply the normalized discounted cumulative gain (*nDCG*) metric. The metric compares the reranked recommendation results from

[1] http://zbw.eu/stw/version/latest/about

[2] https://datorium.gesis.org/xmlui/handle/10.7802/1224

the Paragraph Vectors method (DCG), with the user assessments of the recommendation results which have been generated by the CF-IDF and HCF-IDF profiling methods as a gold standard. To illustrate this, let D be the set of documents, $rel(d)$ is a function that returns one if the document is rated relevant by the user, and zero otherwise. Thus, the normalized discounted cumulative gain $nDCG$ can be computed as:

$$nDCG_k = \frac{DCG_k}{IDCG_k}, \text{ where } DCG_k = rel(1) + \sum_{i=2}^{k} \frac{rel(i)}{\log(i)}$$

and $IDCG_k$ is the optimal ranking computed from the gold standard.

4.3 Results

In Table 1, we present the results of reranking recommendation approach using Deep Learning. As mentioned above, we used two Paragraph Vectors models for reranking the recommended scientific publication. The first model was trained with Wikipedia english language documents, we call it PV-wiki. The second model was trained with ZBW's economics dataset, we call it PV-economics. The results shows that reranking with Paragraph Vectors models can, in most cases, be utilized to improve the recommendation. In addition, the results show that the domain of the training dataset has an effect on the results. This could happen due to the problem of "out of vocabulary words", which occurs when the embedding models are trained on a small dataset (in terms of the number of words) from different domains.

Tab. 1: Normalized discount cumulative gain (nDCG) of the CF-IDF and HCF-IDF plain recommendation results vs. the reranked results of the Paragraph Vectors models (PV-wiki and PV-economics)

	Strategy	Reranking Strategy	nDCG@5 (SD)
Full text	CF-IDF	PV-economics	**0.822 (0.267)**
		PV-wiki	0.811 (0.280)
		None	0.796 (0.274)
	HCF-IDF	PV-economics	0.740 (0.325)
		PV-wiki	0.757 (0.332)
		None	**0.773 (0.327)**
Titles	CF-IDF	PV-economics	**0.671 (0.345)**
		PV-wiki	0.662 (0.342)
		None	0.654 (0.335)
	HCF-IDF	PV-economics	**0.762 (0.332)**
		PV-wiki	0.759 (0.337)
		None	0.734 (0.327)

5 Discussion and Conclusions

We have presented a reranking-based recommender system using Paragraph Vectors. We use CF-IDF and HCF-IDF as initial ranking methods as they had already been proven to provide best recommendations in our previous study.

We have trained two Paragraph Vectors models and utilized them for calculating the vector representation of scientific publications and users' tweets. Using cosine similarity, we compute the scores for reranking the documents. The results show that the Paragraph Vectors reranking method can improve upon previous methods. In the experiments, we employed only two different parameter sets for training two Paragraph Vectors models and focused primarily on reranking the recommendation results from CF-IDF and CF-IDF. In the future, we aim to asses the influence of the training parameters on the performance of our reranking-based recommender system.

Acknowledgement This work was supported by the EU's Horizon 2020 programme under grant agreement H2020-693092 MOVING.

References

[Ch10] Chen, Jilin; Nairn, Rowan; Nelson, Les; Bernstein, Michael; Chi, Ed: Short and tweet: experiments on recommending content from information streams. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, pp. 1185–1194, 2010.

[Go11] Goossen, Frank; IJntema, Wouter; Frasincar, Flavius; Hogenboom, Frederik; Kaymak, Uzay: News personalization using the CF-IDF semantic recommender. In: Proceedings of the International Conference on Web Intelligence, Mining and Semantics. ACM, p. 10, 2011.

[Ka14] Kapanipathi, Pavan; Jain, Prateek; Venkataramani, Chitra; Sheth, Amit: User interests identification on twitter using a hierarchical knowledge base. In: European Semantic Web Conference. Springer, pp. 99–113, 2014.

[LM14] Le, Quoc V; Mikolov, Tomas: Distributed Representations of Sentences and Documents. In: ICML. volume 14, pp. 1188–1196, 2014.

[Mi13] Mikolov, Tomas; Chen, Kai; Corrado, Greg; Dean, Jeffrey: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

[NS16] Nishioka, Chifumi; Scherp, Ansgar: Profiling vs. time vs. content: What does matter for top-k publication recommendation based on Twitter profiles? In: Digital Libraries (JCDL), 2016 IEEE/ACM Joint Conference on. IEEE, pp. 171–180, 2016.