Indian Institute of Technology Kharagpur
Department of Computer Science and Engineering

# SCIENTIFIC PAPER RECOMMENDATION SYSTEM

Riya Bubna - 13CS10041

*Under the guidance of*
Pabitra Mitra

Submitted in part fulfillment of the requirements for the degree of
Bachelor in Technology in Computer Science of IIT Kharagpur, November 2016

# Abstract

This report presents a deep semantic similarity model (DSSM), a special type of deep neural networks designed for text analysis, for recommending target research papers to be of interest to a user based on his research interests. An author's research interests are found using the papers he has written and the papers that he has cited in his publications. The dataset consists of over a million papers, each entry provides the information of the authors, title, abstract and the citations of the paper. A neural network (more specifically, a DSSM) is trained with this dataset and hence, for any unseen paper as input, it decides if the paper should be recommended or not.

# Declaration

This is to certify that:

- The work contained in this thesis is original and has been done by myself under the general supervision of my supervisor.
- The work has not been submitted to any other Institute for any degree or diploma.
- I have followed the guidelines provided by the Institute in writing the thesis.
- I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
- Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.
- Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them in the text of the thesis and giving their details in the references.

Riya Bubna
Department of Computer Science and Engineering
IIT Kharagpur

Date: 8th November, 2016

# Contents

# Chapter 1

# Deep structured semantic model for scientific paper recommendation

## 1.1 Introduction

A recommender system[4] can follow the steps of its user, observe the interests of a group of similar users, and pick items that best suit the user based on either items the user liked (content-based filtering) or implicit observations of the users followers/friends who have similar tastes (collaborative filtering). In the majority of these approaches, the successful match of the recommended item is measured by its utility, usually given a numerical rating by the user based on how much he or she liked the item, a single-dimensional RS. However, users preference for an item may be influenced by one or many contexts. For instance, say a user is looking for a movie that is suitable for a fun family activity, such as a family-friendly movie. Contexts considered in a RS would vary depending on the applications (e.g., movies, books, music, education, etc.) and tasks the system intends to support.

Recommender systems for research articles are useful applications, which for instance help researchers keep track of their research field.

## 1.2 Literature Survey

### 1.2.1 Recommendation System

Recommendation systems are a subclass of information filtering system that seek to predict the "rating" or "preference" that a user would give to an item. Recommender systems have become extremely common in recent years, and are utilized in a variety of areas: some popular applications include movies, music, news, books, research articles, search queries, social tags, and products in general. There are also recommender systems for experts, collaborators, jokes, restaurants, garments, financial services, life insurance and Twitter pages.
The vast growth of information on the Internet as well as number of visitors to websites add some key challenges to recommender systems. These are: producing accurate recommendation, handling many recommendations efficiently and coping with the vast growth of number of participants in the system. Therefore, new recommender system technologies[5] are needed that can quickly produce high quality recommendations even for huge data sets. There are two basic architectures for a recommendation system:

- *Content-Based systems* focus on properties of items. Similarity of items is determined by measuring the similarity in their properties.
- *Collaborative-Filtering systems* focus on the relationship between users and items. Similarity of items is determined by the similarity of the ratings of those items by the users who have rated both items.

**Content-Based Recommendations**

In a content-based system[6], we must construct for each item a profile, which is a record or collection of records representing important characteristics of that item. In simple cases, the profile consists of some characteristics of the item that are easily discovered. For any document, words that characterize the document are computed(using TF-IDF values) and they may be considered as the features.
The ultimate goal for content-based recommendation is to create both an item profile consisting of feature-value pairs and a user profile summarizing the preferences of the user, based on their

row of the utility matrix. We can then estimate the degree to which a user would prefer an item by computing the cosine distance between the users and items vectors.

### Collaborative-Filtering Recommendations

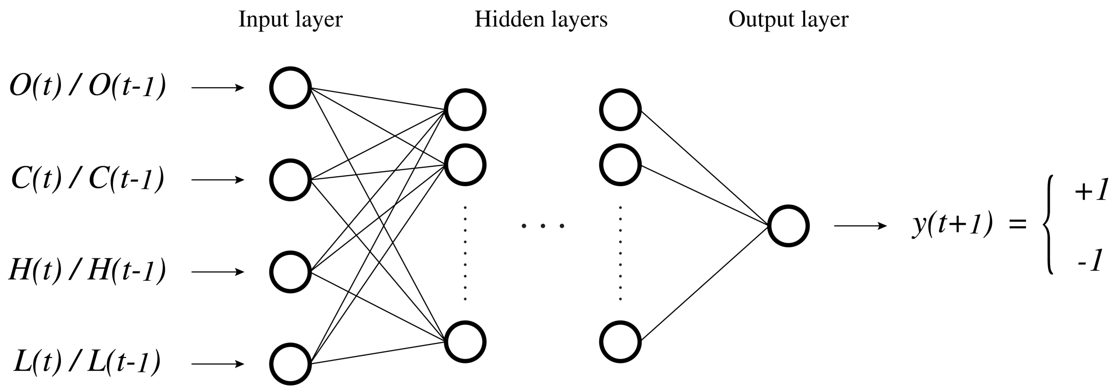
Instead of creating the item-profile vector for an item as mentioned above, we use its column in the utility matrix. Instead of contriving a profile vector for users, we represent them by their rows in the utility matrix. Users are similar if their vectors are close according to some distance measure such as Jaccard or cosine distance. Recommendation for a user U is then made by looking at the users that are most similar to U in this sense, and recommending items that these users like. The process of identifying similar users and recommending what similar users like is called collaborative filtering.

It is hard to detect similarity among either items or users, because we have little information about user-item pairs in the sparse utility matrix. One way of dealing with this pitfall is to cluster items and/or users. Having clustered items to an extent, we can revise the utility matrix so the columns represent clusters of items, and the entry for user U and cluster C is the average rating that U gave to those members of cluster C that U did rate. This process can be repeated several times if we like. Once we have clustered the users and/or items to the desired extent and computed the cluster-cluster utility matrix, we can estimate entries in the original utility matrix.

There are also various modern recommendation approaches such as context-aware approaches, Semantic based approaches, cross-domain based approaches, peer-to-peer approaches and cross-lingual approaches etc. Collaborative filtering in the domain of research paper recommendation is criticized for various reasons. Some authors claim that collaborative filtering would be ineffective in domains where more items than users exist. Others believe that users would be unwilling to spend time for explicitly rating research papers. In general, collaborative filtering has to cope with the possibility of manipulation. Another drawback is that a critical mass of ratings and users is required to receive useful recommendations.

### 1.2.2 Deep Learning

Deep learning is a branch of machine learning based on a set of algorithms that attempt to model high level abstractions in data by using a deep graph with multiple processing layers, composed of multiple linear and non-linear transformations. Research in this area attempts to make better representations and create models to learn these representations from large-scale unlabeled data. Some of the representations are inspired by advances in neuroscience and are loosely based on interpretation of information processing and communication patterns in a nervous system, such as neural coding which attempts to define a relationship between various stimuli and associated neuronal responses in the brain.




Depending on how the architectures and techniques are intended for use, e.g., synthesis/generation or recognition/classification, one can broadly categorize most of the work in this area into three major classes:

- *Deep networks for unsupervised or generative learning,* which are intended to capture high-order correlation of the observed or visible data for pattern analysis or synthesis purposes when no information about target class labels is available.
- *Deep networks for supervised learning,* which are intended to directly provide discriminative power for pattern classification purposes, often by characterizing the posterior distributions of classes conditioned on the visible data.
- *Hybrid deep networks,* where the goal is discrimination which is assisted, often in a significant way, with the outcomes of generative or unsupervised deep networks. This can be accomplished by better optimization or regularization of the unsupervised deep networks.

### Deep Neural Network

A deep neural network(DNN)[1] is an artificial neural network (ANN) with multiple hidden layers of units between the input and output layers. Similar to shallow ANNs, DNNs can model complex non-linear relationships. DNN architectures, e.g., for object detection and parsing, generate compositional models where the object is expressed as a layered composition of image primitives. The extra layers enable composition of features from lower layers, giving the potential of modeling complex data with fewer units than a similarly performing shallow network.

A DNN can be discriminatively trained with the standard backpropagation algorithm. When later layers in the network are learning well, early layers often get stuck during training, learning almost nothing at all. The opposite phenomenon can also occur: the early layers may be learning well, but later layers can become stuck. In fact, there's an intrinsic instability associated to learning by gradient descent in deep, many-layer neural networks. This instability tends to result in either the early or the later layers getting stuck during training.

### Relevance to Recommendation Systems

Generalized linear models with nonlinear feature transformations are widely used for large-scale regression and classification problems with sparse inputs. Memorization of feature interactions through a wide set of cross-product feature transformations are effective and interpretable, while generalization requires more feature engineering effort. With less feature engineering, deep neural networks can generalize better to unseen feature combinations through low-dimensional dense embeddings learned for the sparse features.

Most recommender systems rely on collaborative filtering[7], suffering from the cold start problem where it fails when no usage data is available. Thus, collaborative filtering is not effective for recommending new and unpopular songs. Deep learning methods power the latent factor model for recommendation, which predicts the latent factors from music audio when they cannot be obtained from usage data. A traditional approach using a bag-of-words representation of the audio signals is compared with deep CNNs with rigorous evaluation made. The results show highly sensible recommendations produced by the predicted latent factors using deep CNNs.

# Chapter 2

# Proposed Methodology

## 2.1 Deep Structured Semantic Models

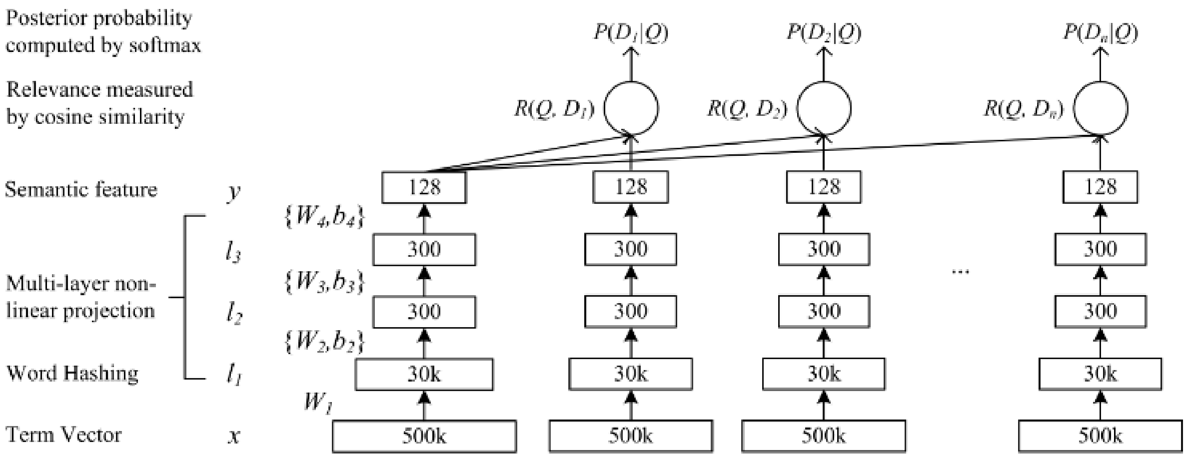



**Figure 1:** Illustration of the DSSM. It uses a DNN to map high-dimensional sparse text features into low-dimensional dense features in a semantic space. The first hidden layer, with 30k units, accomplishes word hashing. The word-hashed features are then projected through multiple layers of non-linear projections. The final layer's neural activities in this DNN form the feature in the semantic space.

The typical architecture of the DSSM[2] is shown above. The input (raw text features) to the DNN is a high dimensional term vector, e.g., raw counts of terms in a query or a document without normalization. Then the DSSM passes its input through two neural networks, one for each of the two different inputs, respectively, and maps them into semantic vectors in a shared semantic space. For Web document ranking[3], DSSM computes the relevance score between a query and a document as the cosine similarity of their corresponding semantic vectors, and ranks documents by their similarity scores to the query.

More formally, if we denote x as the input term vector, y as the output vector, $l_i$, i = 1,...,N-1, as the intermediate hidden layers, $W_i$ as the i-th weight matrix, and $b_i$ as the i-th bias term, we have:

$$l_1 = W_1 x$$

$$l_i = f(W_i l_{i-1} + b_i), i = 2, 3.....N-1$$

$$y = f(W_N l_{N-1} + b_N)$$

where we use the tanh function as the activation function at the output layer and the hidden layers $l_i$, i = 2,...N-1:

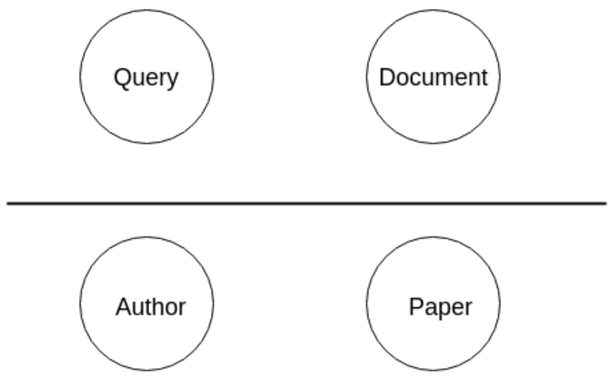
$$f(x) = \frac{1 - e^{-2x}}{1 + e^{2x}}$$

The semantic relevance score between a query Q and a document D is then measured as:

$$R(Q, D) = cosine(y_Q, y_D) = \frac{{y_Q}^T y_D}{||y_Q||.||y_D||}$$

## 2.2 DSSM for Recommendation System

### Mapping DSSM for Recommendation System



# Chapter 3

# Conclusion

## 3.1 Progress Made

The entire dataset was initially cleaned and formatted and stored to a database, having over 2 million entries. The database has 3 tables, each stores different relevant information about the research papers: title, abstract(optional), author(s) and the citation information. This database will be used to extract all data in the project as querying a database is easier and time efficient.

A DSSM model is being implemented currently. DSSM is learned in two stages. First, a stack of generative models are learned to map layer-by-layer a term vector representation of a document to a low-dimensional semantic concept vector. Second, the model parameters are fine-tuned so as to minimize the cross entropy error between the original term vector of the document and the reconstructed term vector. The intermediate layer activations are used as features (i.e.bottleneck) for paper recommendation. A DSSM with 300 units in the first and in the second layers, and 128 units in the output layer is built. The training will be done over 100 epochs, with a learning rate of 0.1. The input dataset has generated separately.

## 3.2 Future Work

- We are going to incorporate community analysis in the DSSM model. A network is said to have community structure if the nodes of the network can be easily grouped into (potentially overlapping) sets of nodes such that each set of nodes is densely connected internally. The community structure and provides a useful visualizing technique. In our case, all authors with similar research interests can be grouped into one single community. Furthermore, research papers can form communities too. Analyzing these communities and taking them into account while recommending might give better results.

- We will use zero shot learning techniques for handling the cold start problem. Zero shot learning in simple terms is a form of extending supervised learning to a setting of solving for example a classification problem when not enough labeled examples are available for all classes. This will help recommend papers to unknown authors and also, papers which aren't related to the papers of the authors in the dataset.

# Bibliography

[1] Deng, Li, Geoffrey Hinton, and Brian Kingsbury. "New types of deep neural network learning for speech recognition and related applications: An overview." 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013.

[2] Song, Xinying, et al. Unsupervised learning of word semantic embedding using the deep structured semantic model. Tech. Rep. MSR-TR-2014-109, 2014.

[3] Huang, Po-Sen, et al. "Learning deep structured semantic models for web search using clickthrough data." Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. ACM, 2013.

[4] Resnick, Paul, and Hal R. Varian. "Recommender systems." Communications of the ACM 40.3 (1997): 56-58.

[5] Adomavicius, Gediminas, and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." IEEE transactions on knowledge and data engineering 17.6 (2005): 734-749.

[6] Lops, Pasquale, Marco De Gemmis, and Giovanni Semeraro. "Content-based recommender systems: State of the art and trends." Recommender systems handbook. Springer US, 2011. 73-105.

[7] Herlocker, Jonathan L., et al. "Evaluating collaborative filtering recommender systems." ACM Transactions on Information Systems (TOIS) 22.1 (2004): 5-53.