Indian Statistical Institute
Kolkata

M.Tech. (Computer Science) Dissertation

Recommender System for Bibliographic Citations

A dissertation submitted in partial fulfillment of the requirements
for the award of Master of Technology
in
Computer Science

Author:
Jayadev Dasika
Roll No: MTC 1213

Supervisor:
Dr. Mandar Mitra
Computer Vision and
Pattern Recognition Unit

M.TECH(CS) DISSERTATION THESIS COMPLETION CERTIFICATE

Student : Jayadev Dasika (MTC1213)
Topic : Recommender System for Bibliographic Citations
Supervisor : Dr. Mandar Mitra

This is to certify that the thesis titled "Recommender System for Bibliographic Citations" submitted by Jayadev Dasika in partial fulfillment for the award of the degree of Master of Technology is a bonafide record of work carried out by him under our supervision. The thesis has fulfilled all the requirements as per the regulations of this Institute and, in our opinion, has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other university for the award of any degree or diploma.

Mandar Mitra

Date : 11^{th} *July*, 2014

Acknowledgements

I would like to thank my dissertation advisor Dr. Mandar Mitra for suggesting this topic to me in the first place, and for helping me with the challenges I have faced.

I would also like to thank all my classmates for always being willing to hold a discussion with me. Their timely help facilitated me in understanding many concepts quicker and better, and the lively discussions have always led to the genesis of new ideas.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

While writing research papers, we wish to find the best possible references for what we have written in our paper. Finding them manually is both time consuming and difficult. A citation recommender system takes a research paper draft as input and outputs citation recommendations. The recommender's job is challenging as the recommendation should not only be relevant to the paper in general, but also should be relevant to the local context of the paper in composition.

1.1 Introduction to Recommender Systems

Recommender systems is one of the fields which has grown in parallel to the web. It is also a field that grew out of necessity, as the amount of information available on the web has become increasingly enormous. John Naisbitt once said: "We are drowning in information but starved for knowledge."[9] So, it is important to have good technologies that can translate information to knowledge. One such technology that has become successful is Recommender Systems. M. Deshpande and G. Karypis defined Recommender Systems as: "a personalized information filtering technology used to either predict whether a particular user will like a particular item (prediction problem) or to identify a set of N items that will be of interest to a certain user (top-N recommendation problem)"[3]

There are many approaches to build Recommender Systems. These approaches are typically classified as follows :

- **Content-based** : Recommendations are selected based on the target user's previously liked *content*.

- **Collaborative Filtering** : Recommendations are selected based on items liked by other users with similar tastes and preferences.
- **Hybrid approaches** : They combine Collaborative Filtering and Content Based Methods.

1.2 Introduction to Citation recommender systems

Current citation recommender systems can broadly be classified into three categories.

- The first category of recommenders try to complete the citation list of an input text. Here, some of the citations are already specified by the author. For example, McNee et al proposed an approach using collaborative filtering that falls into this category. Their algorithm analyses the citation graph and builds ratings. The details of this algorithm are discussed in the next chapter[10].
- The second category of recommenders receive just a text as input and generate recommendations from them. For example, Strohman et al. used a two-step recommendation algorithm. They first generated a candidate list of recommendations using the content and citation graph and in the second step, they ranked these recommendations[14].
- The third category of recommenders, placeholders, ie places where citations should be added, are also specified in the text. For example, He et al proposed an approach which proposed recommendations for specified locations[4].

Our recommender falls into the third category.

Chapter 2

Related Work

2.1 Collaborative Filtering

Collaborative filtering has been the most popular technique used in recommender systems. Collaborative filtering finds similar users and uses this information to make recommendations. Being the most popular technique, collaborative filtering based methods have been tried for the citation recommendation problem. A citation graph is generally used in these methods. This graph is formed using the citations between papers. By following the citation graph for a paper, one can find what papers cite it and what papers are cited by it.
Standard Collaborative Filtering algorithms view dataset as a ratings matrix. In the standard Collaborative Filtering environment, columns represent 'items' and rows represent 'users'. Each entry of the matrix is user's rating for a specific item. Collaborative Filtering algorithms make recommendations by trying to predict what can appear in the blank entries of this matrix. There are several ways to create this ratings matrix from the citation graph.

- The first approach does not use the citation graph. Citations are 'items' and 'users' are actual people (researchers) who rate the citations.
- In the second approach, paper authors are 'users' and citations are 'items'. An author votes for all papers he has cited. This method has been explored by Kautz et al.[5] and Newman[12]. This approach would have problems if the the same author has worked in multiple domains.
- The third and the most popular approach was introduced by McNee et al[10]. Here, a paper is the 'user' and the citation is an 'item'. A paper votes for citations in its reference list.

- Some other methods have also been explored. For instance, both 'items' and 'users' are citations, and the matrix entries correspond to a measure of co-citation. A co-citation metric counts the number of times both citations have occurred together in a single reference list.

2.1.1 Collaborative Filtering Algorithms

Once the ratings matrix is ready, there are many algorithms that can be used to make recommendations using it. The following are some of the popular Collaborative Filtering algorithms :

- *Naive Bayesian Classifier :* It calculates probabilities that a citation in the dataset is related to input and recommends using these probabilities.
- *User-Item CF :* User-Item CF algorithm finds the most similar rows and recommends items of these row.
- *Item-Item CF :* Item-Item CF algorithm finds most similar columns and uses them for recommendation.

2.2 Content-Based methods

Content Based methods use the content of the input paper draft. They try to find papers with similar content and output the corresponding citation as recommendation. The following are the general steps performed by a Content-Based method :

- *Content Analysis :* This is an important step especially when the data is unstructered(e.g text). So, the content should be pre-proccesed to filter out the irrelevant stuff. This step is responsible to translate the unstructered data to a structured form so that the following steps can use the structured form.
- *User Profile Learner :* This step collects data for a particular user, tries to generalize it and build a user profile for each user.
- *Recommeder :* The final step, the recommender uses user profiles build in previous step and makes recommendations.

For example, Strohman et al proposed a content based recommendation algorithm[14]. Its a two-step process. In the first step, 100 papers having similar content are retrieved and added to a set, say S. Then, all citations of these papers are added to S. A target size for S is fixed and this process is

repeated iteratively. Now, in the next step, items in S are ranked using some features like Text Similarity, Same Author etc and the top items are output as citation recommendations.

Most content-based methods use words as features. TF-IDF is the most popular weighing scheme used. The most popular method to store the item representation is the Vector Space model.

Research papers have various fields from which words can be extracted. These fields include title, abstarct, inroduction, author provided keywords, bibliography apart from the paper's body text. It is natural to assume that words occuring in different fields have different importance. Words in title maybe more meaningful than words in a text. Nascimento et al. accounted for this and weighted terms from the title three times more than terms from the body-text, and text from the abstract twice as much [11]. In addition, data obtained from external sources like citations may also be used in content-based methods. We use citation context in our work which is described in a later scetion.

The biggest challenge for all content-based methods, as we have also experienced, is the dependency on access to a huge corpus of research papers. For research papers, content access is not trivial. PDFs must be processed and converted to text, fields must be identified, and features such as terms and citations must be extracted. None of these tasks are trivial. For instance, Beel et al used the heuristic that the largest text on the first page of a PDF is its title[1]. They report an accuracy of only about 70%.

Chapter 3

Our work

3.1 Problem Definition

Let d be a document and D be the document corpus. In a document, the local context of a citation is the text surrounding the placeholder for the citation. He at al defined local recommendation as :

Definition Given a context of citation c with respect to document d, a local recommendation is a ranked list of citations in a corpus D that are recommended as candidates for the placeholder associated with c [4]

Given an input document with placeholders, our aim is to make local recommendations for each placeholder in the document.

3.2 Methodology

This section explains the methodology used, the implementation details will be presented in a later section. The input given is a paper draft with placeholders. First, the document is pre-processed. During pre-processing, the document is tokenized by space characters, all upper case letters are converted to lower case, punctuations are removed, stopwords are removed, and finally words of small length are discarded. The remaining words are stemmed. Then, for each citation placeholder, the words surrounding the placeholder are collected. These words form the context for the placeholder. Word histogram of each context are constructed and used as the feature vector corresponding to the context. These word histograms are used to calculate similarity between contexts. To measure the similarity between two contexts, firstly the intersection between the two contexts is obtained. In case of multiple occurence of a word in the contexts, the minimum frequency of the word is taken in the intersection i.e if a word $x's$ frequency is n and m

in two contexts, the intersection of these two contexts would have the word x with frequency $min(n, m)$. Then, similarity is calculated where the words are weighted using the TF-IDF weighting scheme

$$Similarity(x, y) = \sum_{w \in Intr(x,y)} TF(w) * IDF(w) \tag{3.1}$$

TF(w), the term frequency of the word w, is obtained using the formula :

$$TF(w) = log(1 + f(w)) \tag{3.2}$$

where $f(w)$, the frequency of word w, is obtained from the word histogram of the intersection of contexts

IDF, the inverse document frequency is obtained using the formula :

$$IDF(w) = log(\frac{N}{1 + df(w)}) \tag{3.3}$$

where $df(w)$, the document frequency of word w is the number of documents in the corpus that contain w and N is the total number of documents in the corpus. To calculate IDF, the data provided by Google Inc., which contains English word n-grams and their observed frequency counts, is used[2].

These contexts are considered as input for the learning algorithm. Each cited paper is a class label in the learning algorithm. Multi-class classification is used to learn the class labels for the given input feature vector. A k-class(k is the total number of cited papers in the corpus) classification problem has a labelled training sample $\{(x_1, y_1), ..., (x_m, y_m)\}$, where x_i's are the feature vectors and y_i are the corresponding class labels, $y_i \in \{1, 2, ..., k\}$, and a unlabelled testing sample $\{x_1^*, ..., x_l^*\}$ of feature vectors. Three multi-class classification algorithms are used :

- **Nearest neighbour** : An input is assigned to the class of its nearest neighbour calculated using the similarity measure given in equation (3.1). Since classes are papers in our context, the paper corresponding to the nearest context is the first citation recommendation. The paper corresponding to the second nearest neighbour is the second citation recommendation and so on.

- **Nearest mean** : This is similar to nearest neighbour except that mean similarity is used as the similarity metric. Mean similarity of a context with a class is the mean of the similarity obtained with respect to individual contexts of that class.

- **Cluster and Classify** : This algorithm was proposed for spam detection by Antonia et al.[8] It has three steps:

  - **Clustering step** : For the clustering step of the algorithm, we consider both the training and the test samples. Only the feature vectors from the training sample are used for this step, the class labels are not used. $k-means$ clustering algorithm is used. The details of $k-means$ clustering algorithm can be found in [13]. Results from [7] showed that the best performance is obtained when number of clusters equalled the number of pre-defined classes and so we have fixed k in $k-means$ algorithm to be equal to the number of cited papers in the corpus (which is the number of classes).

  - **Expansion step** : Each cluster obtained contributed one meta-feature to the feature space of training and test sets, i.e. k meta features are created. If a sample belonged to i^{th} cluster, then its i^{th} meta-feature was 1 and the remaining meta-features were 0.

  - **Classification step** : Nearest neighbour classifier is used. The meta-features are weighed by applying TF-IDF scheme to the clusters. For all samples in cluster i, the TF of the i^{th} meta-feature equalled 1 i.e. the sample contributed a frequency of 1 to the cluster. The document frequency (df) is equal to the size of the cluster and N is equal to total number of samples (training+test). IDF is then obtained using the formula (2.3).

It should be pointed out at this juncture that the context of a paper's citation is used as its representative feature instead of its actual content because in many cases, the actual contents of the paper may not be available. The whole process is summarized in the following figure :

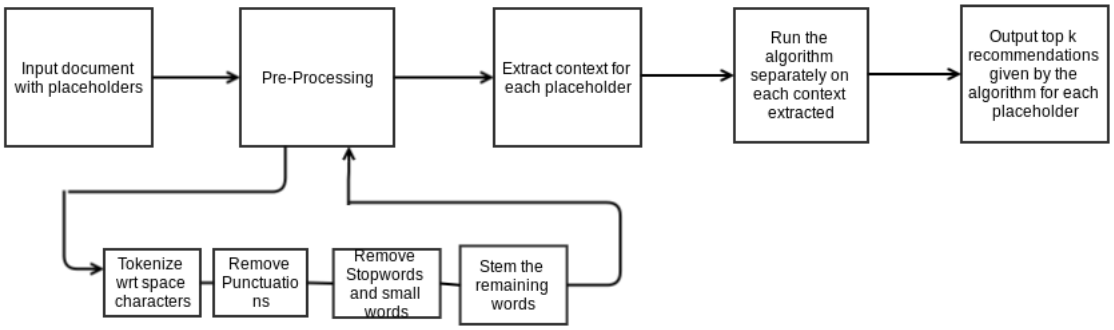

Figure 3.1: Methodology

3.3 Experimental Evaluation

The algorithms presented earlier were coded in Python.

3.3.1 Dataset

An intial set of 152 papers were downloaded from ACM Digital Library from the topic "Recommender Systems".
Each of these papers were pre-proccesed. Stopwords list was obtained from the corpus of nltk package in Python. Punctuation list was obtained from the package string. The PorterStemmer class available with the nltk package was used for stemming. All words of length less than three (before stemming) were discarded.
Then, for each of the citations in the paper, the citation context was collected. The results in [6] indicated that fifty words around the citation best represented the context. So, fifty words around each citation (25 before and 25 after) were used as the citation context. This process of collecting citation contexts was repeated for each of the papers in the collection. An inverted index was used to store them where the citation contexts were stored as a list indexed by the cited paper. Each cited paper was uniquely represented by its bibtex key obtained from Google Scholar. Then, all the papers (and the corresponding contexts) which had less than four corresponding contexts were discarded. 295 papers survived this filtering. In all, there were 1702 contexts across these papers.

3.3.2 Results

A four-fold cross-validation technique was used. The whole data was divided randomly into four sets. Since each cited paper had atleast four corresponding contexts, each cited paper appeared in each of the sets. Three of the sets were used as training samples and the remaining set was used as the test sample. Results were collected by running each of the algorithms four times, where the test set changed in every run. The result reported is an average of the results obtained in these runs.

Evaluation Metrics used

Hit Percentage : If the right recommendation was at the top of the list, it was counted a hit, otherwise it was counted a miss.

MRR :

Definition Mean Reciprocal Rank is a statistical measure that is generally used for evaluating any process that produces a ranked list of responses for a query. The reciprocal rank for a query is the multiplicative inverse of the rank of the first correct answer. Mean reciprocal rank is the average of these reciprocal ranks over a set of queries.

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{3.4}$$

MRR is used as the evaluation metric here as the output is a ranked list and there is a single correct answer.

The table below summarizes the results obtained :

	MRR	Hit percentage	Present in top 5 percentage
Nearest Neighbour	0.607(0.586-0.623)	41.09(39.01-43.63)	65.14(63.34-67.45)
Nearest Mean	0.692(0.670-0.714)	47.41(45.91-49.29)	71.43(69.83-72.91)
Cluster and Classify	0.761(0.738-0.786)	46.71(45.67-48.58)	73.87(70.37-75.46)

Table 3.1: Results

Chapter 4

Conclusions

4.1 Result Analysis

Nearest mean classification performed better than the Nearest Neighbour classification in the Hit percentage, MRR score and also in Present in top 5 percentage. The MRR score and Present in top 5 percentage improved on adding the natural cluster information to the feature set but the hit percentage slightly reduced.
It should be noted, however, that the above analysis was performed on a small dataset. Also, the dataset contained papers from the same topic. We need to perform tests on a larger dataset before drawing any inferences.

4.2 Towards a citation recommendation tool

The eventual goal of work in this direction is to provide a tool which can act as an add-on to document editors like Latex where citation recommendations are made to the author of a research paper as he is composing it. The following format conversion codes have been implemented that may later be used in the tool :

- *bbl to bibtex converter :* Takes a .bbl file as input, converts each bibliography entry in that file to Bibtex format and outputs a .bib file.
- *All Citations bibtex extractor :* It takes the paper in text format as input and outputs bibtex for all the cited papers. This code works for paper formats of major journals and conferences.

4.3 Future Work

As said eariler, the goal of work in this direction is a citation recommendation tool. A good amount of preliminary work for this has been done. Here is a list of future work and challenges :

- Use the information provided by the citation graph to make better recommendations.
- Addressing the issues involved in scaling the model to deal with very huge data. The algorithms used till now work fine on a small dataset. As the dataset size increases, a method like nearest neighbour may no longer be feasible.
- Only Content Based algorithms have been used in our work. A Hybrid system, which uses both content based and collaborative filtering techniques may provide better results.
- Extension of current approaches to scenarios where placeholders are not mentioned or where only a partial paper draft is given as input.

Bibliography

[1] Joeran Beel, Stefan Langer, Marcel Genzmehr, and Christoph Müller. Docear's pdf inspector: title extraction from pdf files. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, pages 443–444. ACM, 2013.

[2] T Brants and A Franz. Web 1t 5-gram version 1 (2006). *Linguistic Data Consortium, Philadelphia*.

[3] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[4] Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, and Lee Giles. Context-aware citation recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 421–430. ACM, 2010.

[5] Henry Kautz, Bart Selman, and Mehul Shah. Referral web: combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.

[6] Hema Swetha Koppula. Context aware citation recommendation system.

[7] Antonia Kyriakopoulou and Theodore Kalamboukis. Using clustering to enhance text classification. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 805–806. ACM, 2007.

[8] Antonia Kyriakopoulou and Theodore Kalamboukis. Combining clustering with classification for spam detection in social bookmarking systems. *Proceedings of ECML/PKDD Discovery Challenge 2008 (RSDC 2008)*, pages 47–54, 2008.

[9] Daniel T Larose. *Discovering knowledge in data: an introduction to data mining*. John Wiley & Sons, 2014.

[10] Sean M McNee, Istvan Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K Lam, Al Mamunur Rashid, Joseph A Konstan, and John Riedl. On the recommending of citations for research papers. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 116–125. ACM, 2002.

[11] Cristiano Nascimento, Alberto HF Laender, Altigran S da Silva, and Marcos André Gonçalves. A source independent framework for research paper recommendation. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pages 297–306. ACM, 2011.

[12] Mark EJ Newman. Scientific collaboration networks. i. network construction and fundamental results. *Physical review E*, 64(1):016131, 2001.

[13] Michael Steinbach, George Karypis, Vipin Kumar, et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, 2000.

[14] Trevor Strohman, W Bruce Croft, and David Jensen. Recommending citations for academic papers. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 705–706. ACM, 2007.