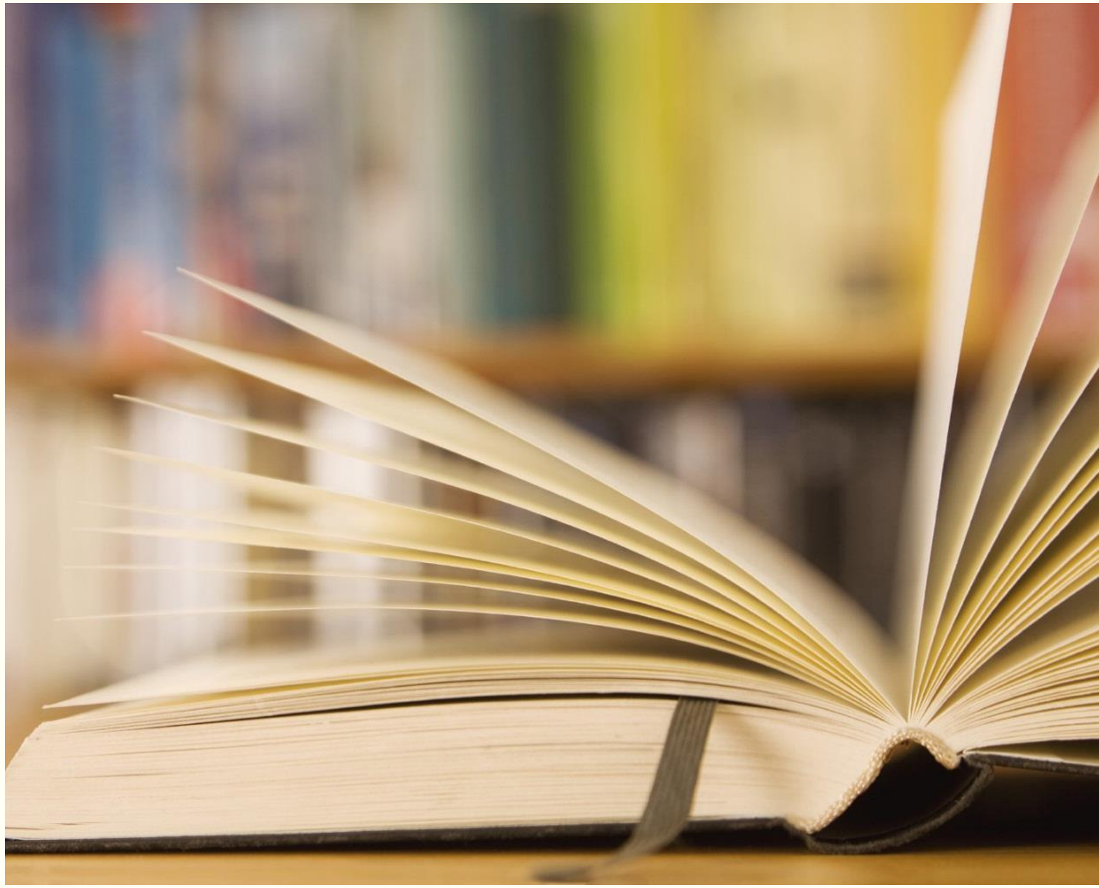
BIBLIOGRAPHIC CITATION RECOMMENDER SYSTEM

P Omkar Ashrit(CS1914)

under the guidance of: Prof. Mandar Mitra



Agenda

- Introduction
- Collection Overview
- Recommendation Approaches
- Results
- Implementation
- Future Scopes

Introduction

- Citation

> the proliferation of new and upcoming journals and conferences, this involves quite a bit of work. According to a recent survey [1] more than 200 research articles have been published on recommender systems for bibliographic citation, starting with the first publication on recommender systems for research citations way back in 1998 [3]. Principally, three approaches

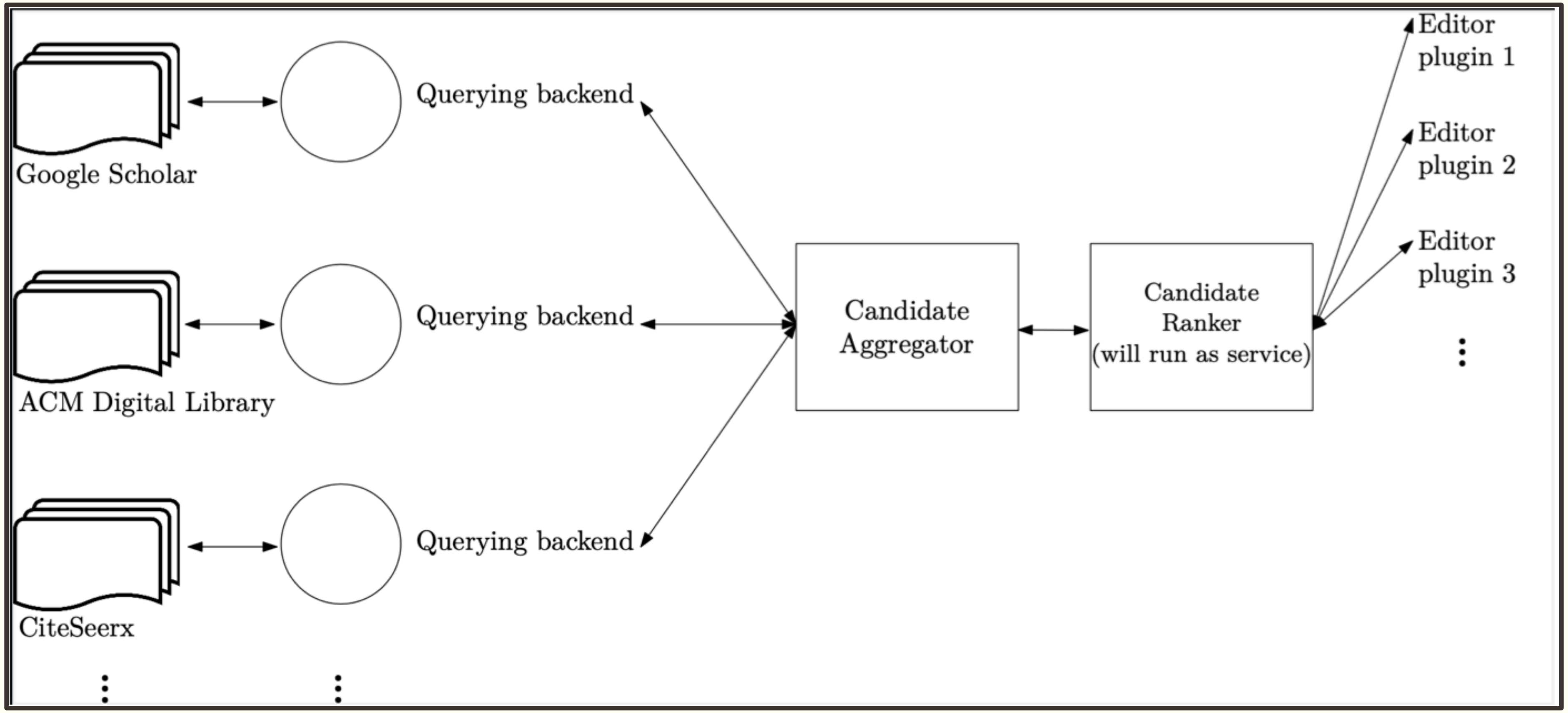
- BCRS
  - A plugin-like tool that is integrated into an editor used for writing academic articles
  - The tool recommends references that may be cited at specific locations in the paper.

Problem Statement

Create a practical system to assist an author who is writing a paper

- Given a partially written paper with placeholders for citations, Recommend papers to be cited from a collection of research papers, suggest papers to cite

Schematic Diagram



Our Contribution

- Reproduce the result by implementing the models and evaluation of methods identified by Kunal Roy
  - Improve the results with better models and techniques

- Compare different approaches to the recommendation problem using a common dataset.

- Implementation of prototype plugin

Collection Overview:

- 630,199 scholarly articles from various sub disciplines from CiteSeerX
- Unique ID for each article: DOI (Digital Object Identifier)
- Different versions of same paper have different DOIs

- Cluster ID: Same for different versions of same paper
- 22,23,307 different cluster IDs

- Citation Context: 400 characters around the citation

Collection Overview: Document XML

XML File:


```xml
<paper>
  <title>Combining Adaptive and Deterministic Routing: Evaluation of a Hybrid Router.</title>
  <author>Dianne R. Kumar</author>
  <author>Walid A. Najjar</author>
  <venue>CANPC</venue>
  <year>1999</year>
  <doi>10.1.1.1.1504</doi>
  <abstract>None</abstract>
 ▼<citations>
   ▶<citation>
     ...
    </citation>
   ▶<citation>
     ...
    </citation>
   ▼<citation>
      <raw>A. Chien. A cost and speed model for k-ary n-cube wormhole routers. In IEEE Proc.
      of Hot Interconnects, Aug. 1993.</raw>
      <contexts>uing delay by providing multiple path options. However, the router delay for
      deterministic routers, and consequently their corresponding clock cycles, can be
      significantly lower than adaptive routers =-=[2, 4]-=- This difference in router delays
      is due to two main reasons: number of VCs and output (OP) channel selection. Two VCs are
      suffcient to avoid deadlock in dimension ordered routing routing the OP channel
      selection policy depends also on the state of the router (i.e the occupancy of various
      VCs) causing increased router complexity and higher router delays. The results reported
      in =-=[2, 4]-=- show that the router delays for adaptive routers are about one and a
      half to more than twice as long as the dimension-order router for wormhole routing. The
      advantage of adaptive routing in reducing reduces blocking. 2.3 Modeling Router Delay In
      this section we describe a router delay model for the virtual cut-though deterministic
      and adaptive routers. The model is based on the ones described in =-=[4, 2, 9]-=-. These
      models account for both the logic complexity of the routers as well as the size of the
      crossbar as determined by the number of VCs that are multiplexed on one PC. These models
      were modi ed to erministic channels. Note that this relationship includes the delivery
      port. Delay equations for the routers are derived, using the above parameters. The
      constants in these equations were obtained in =-=[4]-=- using router designs along with
      gate-level timing estimates based on a 0.8 micron CMOS gate array process. The three
      main operations (delays) prevalent in all of the routers simulated here are as fol rk in
      understanding the e ects of router complexity on cycle time involved deterministic
      routers [7, 5, 10]. Adaptive and deterministic router implementations were then compared
      for worm-hole routing =-=[2, 4, 9]-=-. However, the comparison in [2, 4] does not
      account for the reduced queuing delay in adaptive routing. In [9] the reduction in
      queuing delay for worm-hole routing is taken into account and the compar</contexts>
      <clusterid>352</clusterid>
    </citation>
```


Original Paper:

queuing delay by providing multiple path options.

However, the router delay for deterministic routers, and consequently their corresponding clock cycles, can be significantly lower than adaptive routers [2, 4]. This difference in router delays is due to two main reasons: number of VCs and output (OP) channel selection. Two VCs are sufficient to avoid deadlock in dimension ordered routing [6]; while adaptive routing (as described in [8, 3]) requires a minimum of three VCs in k-ary n-cube networks. In dimension-ordered routing, the OP channel selection policy only depends on information contained in the message header itself. In adaptive routing the OP channel selection policy depends also on the state of the router (i.e the occupancy of various VCs) causing increased router complexity and higher router delays.

The results reported in [2, 4] show that the router delays for adaptive routers are about one and a half to more than twice as long as the dimension-order router for worm-hole routing. The advantage of adaptive routing in reducing queuing delays is evaluated and reported in [9] for worm-hole routing. A typical comparison of deterministic versus adaptive routing message latencies (accounting for the differences in cycle times) is shown in Figure 1: at low traffic and for

Collection Overview: Query XML

XML File:


```
▼<queries>
  ▶<top>
   ...
   </top>
  ▼<top>
     <paper_num> 2 </paper_num>
     <paper_title> Mitigating denial of service attacks: A tutorial. </paper_title>
     <doi> 10.1.1.100.6792 </doi>
     <year> 2005 </year>
     <paper_abstract> This tutorial describes what Denial of Service (DoS) attacks are, how they
     can be carried out in IP networks, and how one can defend against them. Distributed DoS
     (DDoS) attacks are included here as a subset of DoS attacks. A DoS attack has two phases: a
     deployment and an attack phase. A DoS program must first be deployed on one or more
     compromised hosts before an attack is possible. Mitigation of DoS attacks requires thus
     defense mechanisms for both phases. Completely reliable protection against DoS attacks is,
     however, not possible. There will always be vulnerable hosts in the Internet, and many attack
     mechanisms are based on ordinary use of protocols. Defense in depth is thus needed to
     mitigate the effect of DoS attacks. This paper describes shortly many defense mechanisms
     proposed in the literature. The goal is not to implement all possible defenses. Instead, one
     should optimize the trade-off between security costs and acquired benefits in handling the
     most important risks. Mitigation of DoS attacks is thus closely related to risk management.
     </paper_abstract>
     <query_num> 201 </query_num>
     <text> Denial of Service (DoS) attacks have proved to be a serious and permanent threat to
     users, organizations, and infrastructures of the Internet [26]. The primary goal of these
     attacks is to prevent access to a particular resource like a web server [8]. A large number
     of defenses against DoS attacks have been proposed in the literature, but none of them gives
     reliable protection. There will always be vulnerable hosts in the Internet to be used for DoS
     purposes. In addition, it is very difficult to reliably recognize and filter only attack
     traffic without causing any collateral damage to legitimate traffic. This paper describes,
     how DoS attacks can be carried out and how a victim can mitigate them in ordinary IP
     networks. Especially wireless ad hoc networks have their additional vulnerabilities, but
     these kind of wireless networks are not the subject of this paper. </text>
     <query_num> 202 </query_num>
     <text> There are two major reasons making DoS attacks attractive for attackers. The first
     reason is that there are effective automatic tools available for attacking any victim [9],
     i.e., expertise is not necessarily required. The second reason is that it is usually
     impossible to locate an attacker without extensive human interaction [12,59] or without new
     features in most routers of the Internet [11]. </text>
     <query_num> 203 </query_num>
     <text> A DoS attack aims in degrading availability. Denial of Service has been defined as the
     prevention of authorized access to resources or the delaying of time-critical operations
     [22]. Examples of these resources are network bandwidth, processing capacity, disk space,
     memory, and static memory structures [8]. DoS attacks can be classified based on the number
     of sources included in the attack [26]. In a basic DoS attack the attacker uses a single
     source host to send attack traffic to a victim. In a DDoS attack an attacker uses multiple
     source hosts to send attack traffic to one or more victims simultaneously. </text>
     <query_num> 204 </query_num>
```


XML Tags:

- 226 Papers
- queries
  - top
    - paper_num
    - paper_title
    - doi
    - year
    - paper_abstract
    - query_num
    - text
    - query_num
    - text
    - .
    - .
    - .
  - top
    - .
    - .
    - .

Recommendation Approaches

Paper1 – cid_10
- a, cid1
- b, cid2
- c, cid3

Paper2 – cid_20
- d, cid2
- e, cid3
- f, cid4

paper3 – cid_30
- g, cid3
- h, cid4
- i,cid5

Content Based Methods:
- cid_10 : a b c
- cid_20 : d e f
- cid_30 : g h I

Query From:
- Title + Abstract + Context
- Only Context

Reference Directed Indexing:
- cid1 – a
- cid2 – b d
- cid3 – c e g
- cid4 – f h
- cid5 – I

Query From:
- Only Context

Recommendation Approaches: Pre-trained Models

- Avg Word2Vec:
  - Pretrained model released by Google
  - The output vector is a 300-dimensional sparse vector for each word in a sentence
  - Avg Word2Vec: Average of vectors of each word of a sentence that are in vocabulary
  - Trained on part of Google News dataset (about 100 billion words)
  - Contains vectors for 3 million words and phrases
  - Skip gram model

- Universal Sentence Encoder:
  - Pretrained model released by Google
  - The output vector is a 512-dimensional dense vector for each input sentence
  - Trained on Wikipedia, News, Movie and Customer reviews data

Recommendation Approaches: TF-IDF

- SK Learn TF IDF Vectorizer
  - TF: Term Frequency (count of the words present in document from its own vocabulary)
  - IDF: Inverse Document Frequency (importance of the word to each document).

- Takes the corpus as input and learns vocabulary and returns document-term matrix.

- For each input sentence, transforms documents to document-term matrix.

- The output vector is a sparse vector

Results

Indexing Method	Query From	Relevant Returned	MAP	Reciprocal Rank
Content Based	Title + Abstract + Context	207	0.0682	0.1758
Content Based	Context	270	0.1551	0.3821
Reference Directed Indexing	Context	**783**	**0.7549**	**0.9012**
Avg Word2Vec	Context	250	0.1246	0.3503
Universal Sentence Encoder	Context	285	0.1606	0.4010
SK Learn TF-IDF	Context	560	0.3739	0.6463

Implementation: Visual Studio Code

- Visual Studio Code (VS Code) is an open-source cross platform IDE made by Microsoft
- One of the top 3 most downloaded IDEs worldwide. (source)
- Supports extensions for additional functionality

Implementation: Extension

- Back-end:
  - Java HTTP server listening to a specific port
  - Handles POST requests to 3 different URLs
- Front-end:
  - Written in TypeScript(developed by Microsoft) using VS Code API
  - The extension makes POST request to the server with the user query and shows results
- Demo

Future Scope

- Instead of fixed length of characters, full sentences could be taken as context around a citation for the sentence embeddings to work better
- Fine tunning the models with our corpus having vocabulary specific to our use case
- Attaching a live database and deploying the extension for well-known IDEs
- Optimizing the extension to scale and handle higher number of parallel users
- Automate the process of context selection and search with a citation place holder

References

- Caragea C. et al. (2014) CiteSeerx: A Scholarly Big Dataset. In: de Rijke M. et al. (eds) Advances in Information Retrieval. ECIR 2014. Lecture Notes in Computer Science, vol 8416. Springer, Cham. https://doi.org/10.1007/978-3-319-06028-6 26
- Dwaipayan Roy. An Improved Test Collection and Baselines for Bibliographic Citation Recommendation. The 26th ACM International Conference on Information and Knowledge Management (CIKM-2017), Sin- gapore, November 6-10, 2017.
- Dwaipayan Roy, Kunal Ray, Mandar Mitra. From a Scholarly Big Dataset to a Test Collection for Bibliographic Citation Recommendation. The 30th AAAI Conference on Artificial Intelligence (AAAI-16): AAAI Workshop: Scholarly Big Data 2016, pp.705-710, Phoenix, Ari- zona, USA, February 12-17, 2016.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. Scikit-learn: Machine Learning in Python. Journal Of Machine Learning Research. 12 pp. 2825-2830 (2011)
- https://github.com/usnistgov/trec eval
- https://tfhub.dev/google/universal-sentence-encoder/4
- https://code.google.com/archive/p/word2vec/
- https://lucene.apache.org/
- https://code.visualstudio.com/api

Thank you!!