Multimedia Technology

Lecture 4: Miscellaneous Techniques in IR

Lecturer: *Dr*. Wan-Lei Zhao

Autumn Semester 2024

Email: wlzhao@xmu.edu.cn,

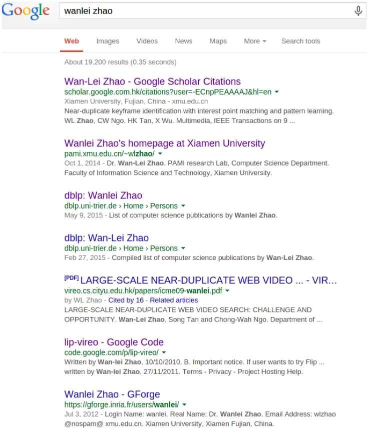
Outline

Pagerank: the motivation (1)

- Retrieval results returned by basic IR system usually are not satisfactory
- There are many reasons behind this
  1. It is actually a very tough issue
  2. Nearly all IR systems face the scalability issue
  3. Users are not able to express what they want by keywords only
  4. The same keyword for different people means different thing, e.g. "apple"
- It requires natural language understanding: **artifical intellegence**
- Hundreds of reranking approaches have proposed to optimize the search results
  - Share the story about SIGIR

Pagerank: the motivation (2)

- Keywords are very few
- Too many pages share similar similarity score



Page hyper-links

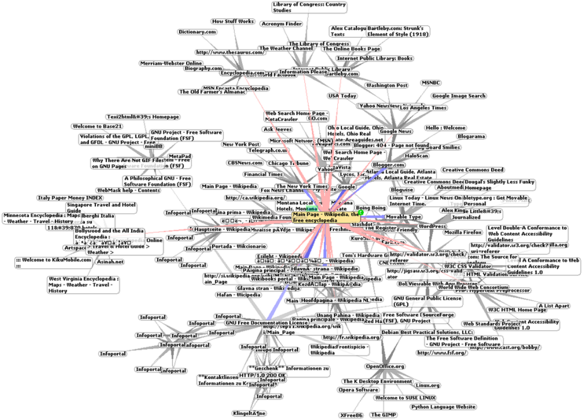
- We are now going to consider
- how hyper-links help to improve the search quality

```html
<html>
<head>page head</head>
<body>
<p>HTML tutorials are available</p>
<a href="http://www.w3schools.com">hyper-link1</a>
<p>WWW standards are available</p>
<a href="http://www.w3.org">hyper-link2</a>
</body>
</html>
```

Pagerank: explained (1)

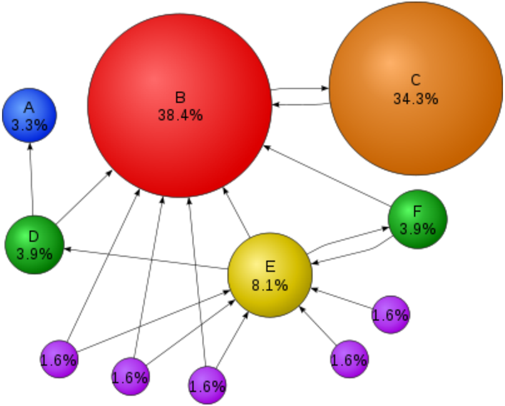
- Pagerank is one of the most successful reranking approaches
- It is a re-ranking approach
- It happens when we have the retrieval results
- Basica idea: make use of the hyperlinks between webpages
  - Pages being linked (pointed to) to by other pages should be important and ranked higher
- Start-up technology for Google

Pagerank: explained (2)



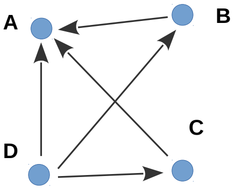
- We are connected by Internet
- Webpages are connected by hyperlinks

Pagerank: explained (3)



- Higher weights (pagerank) are assigned to the pages that have many in-ward links
- Notice that out-ward links will not impact your own ranking

Pagerank: build the model



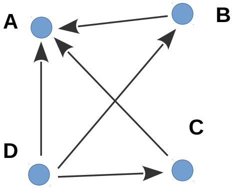
- Given 4 webpages, and the hyperlinks between them
- Calculate pagerank for each of them as following, PR(.) for all the pages are initialized to **0.25**

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}, \tag{1}$$

where PR(.) is the current pagerank,

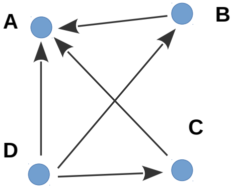
L(.) is num. of out-ward links

Pagerank: build the model



$$
\begin{aligned}
PR(A) = \frac{0.25}{1} + \frac{0.25}{1} + \frac{0.25}{3}, \\
PR(B) = \frac{0.25}{3}, \\
PR(C) = \frac{0.25}{3}, \\
PR(D) = 0
\end{aligned}
$$

Pagerank: the damping factor

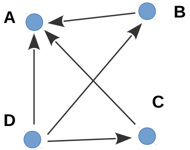


- Given **N** is the num. of webpages, **d** is the damping factor,

$$
\begin{aligned}
PR(A) &= \left(\frac{0.25}{1} + \frac{0.25}{1} + \frac{0.25}{3}\right)\cdot d + \frac{1-d}{N}, \\
PR(B) &= \frac{0.25}{3}\cdot d + \frac{1-d}{N}, \\
PR(C) &= \frac{0.25}{3}\cdot d + \frac{1-d}{N}, \\
PR(D) &= 0\cdot d + \frac{1-d}{N}
\end{aligned}
$$

Pagerank: the procedure

1. Produce Adjacent matrix by collecting all the webpage links
2. Initialize PR(.) to c
3. Do
4. Calculate PR(.) for each webpage
5. Update PR(.) for each webpage
6. Until convergence



$$M = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Pagerank: tricks to promote your webpage

- Share the story about Google
  - What Google means
  - Pagerank is born in the right season
  - Turning point of Google
  - Do we need to **reinvent the wheel**?
- Ask some webpage (has higher pagerank) to link to your webpage
  - Pagerank can be found by install firefox Toolbar or from pagerank website
  - Google robot will ignore hyperlink shares the same color as the background
- Register to Google Webtool
  - Once Google robot visits your site
  - Try to search and click-in your website with Google from different places

Outline

Recap

- We are able to retrieve documents on inverted files
  - Structure of inverted files
  - Static and dynamic inverted files
- We are able to evaluate the performance of IR system
  - Recall, Precision and F-measure
  - mean Average Precision

Puzzle: distributed and centralized Internet

- Internet is the biggest distributed system in the world
  - No central coordinator for information or computing resources
  - Machines, resources, societies are loosely connected
  - The major interface is web browser

- Search engine comes to play a unique role
- Ironically, it is somehow a central coordinator
- Without search engine, we are in dark
- Search engine is the de-facto interface to *WWW*, however not to everything

Web crawler: the information collector

- In June 1993, Matthew Gray from MIT wrote a perl script
- It is able to collect URLs, and keeps tracking on them
- It is also able to identify new websites

- It is the first web robot but not the first search engine
- The idea inspired many programmers to follow-up, leads to the birth of search engine
- Note that only 130 websites in the world in June 1993

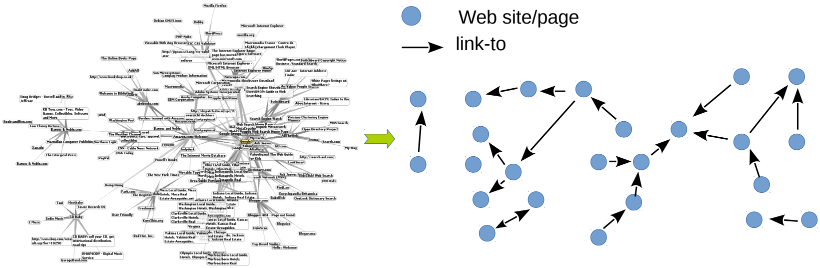
Web crawler: the general steps

- Other names: web robot and web spider
- Feed web robot with several URL seeds, the robot crawls websites into a database for archiving
- General steps:

Crawling (seed pages *S*)

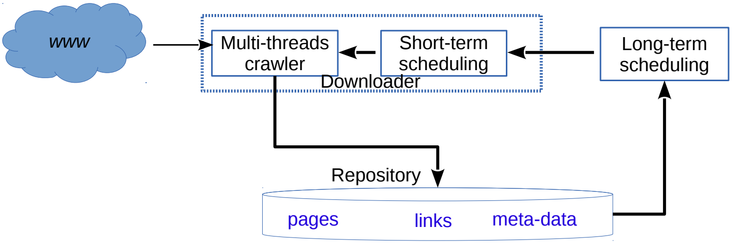
(1) URLQueue ⟵ *S*
(2) ***do*** {
(3) *p* ⟵ *Select-URL*(URLQueue)
(4) content ⟵ *Download*(p)
(5) (text, links, structure, ...) ⟵ Parse(content)
(6) URLQueue ⟵ Add-new-links(*URLQueue*, *links*)
(7) } ***until*** (terminate condition)

Web crawler: the model



- It is a graph transverse problem
- Theoretically speaking, all nodes (pages) must be visited
- Either depth first or width first is fine
- The links between sites will be captivated later for ranking

Web crawler: the framework



- Crawling is a time consuming task

Web crawler: duties

1. Parsing DNS
   - DNS maintains the map between URL and IP
   - Frequent interaction with DNS causes overload
   - Caching DNS record is necessary
2. Normalize URL
   - Same site might be written in different way
   - "yahoo.com.cn" and "yahoo.cn"
3. Parsing web pages
   - HTML mark-ups have no semantic meaning
   - However, they indicate the structure of the page
4. Exceptions handling: soft **404 Error** page
5. Handling duplicate pages, partial duplicate rate: 29%; full duplicate rate: 22%

Web crawler scheduling: an example

- Example of ‘sitemap.xml’

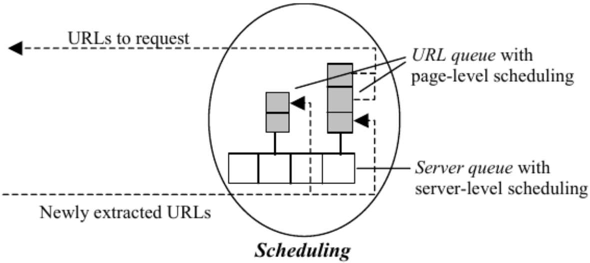
```
▼<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  ▼<url>
     <loc>http://cmmlab.xmu.edu.cn</loc>
     <lastmod>202-09-22</lastmod>
   </url>
  ▼<url>
     <loc>http://cmmlab.xmu.edu.cn/wlzhao.htm</loc>
     <lastmod>202-09-22</lastmod>
   </url>
  ▼<url>
     <loc>http://cmmlab.xmu.edu.cn/wlzhao_cn.htm</loc>
     <lastmod>202-09-22</lastmod>
   </url>
  ▼<url>
     <loc>http://cmmlab.xmu.edu.cn/resc.html</loc>
     <lastmod>202-09-22</lastmod>
   </url>
  ▼<url>
     <loc>http://cmmlab.xmu.edu.cn/research.html</loc>
     <lastmod>202-09-22</lastmod>
   </url>
  ▼<url>
     <loc>http://cmmlab.xmu.edu.cn/pub.html</loc>
     <lastmod>202-09-22</lastmod>
   </url>
  ▼<url>
     <loc>http://cmmlab.xmu.edu.cn/members.html</loc>
     <lastmod>202-09-22</lastmod>
   </url>
 </urlset>
```

Web crawler: scheduling strategies (1)

1. Intuitively, hottest sites should be crawled frequently
   - For example, sohu.com should be crawled in every 30 minutes
2. Depth-first? or breadth-first?
3. Page quality should be considered
   - Allocate more computing resources to these high quality pages
   - These pages are more meaningful to users too

Web crawler: scheduling strategies (2)

1. An exemplar framework



- Scheduling takes place on two levels: server and URLs
- Server queue and URL queue have been built

Web crawler: scheduling strategies (3)

- Three typical stategies

1. Breadth-first scheduling
   - First-in-first-out principle
2. Performance based scheduling

$$R(s,i) = \frac{P(s,i)}{T(s,i)} \tag{2}$$

where P(s,i) is the numb. of pages from ith server

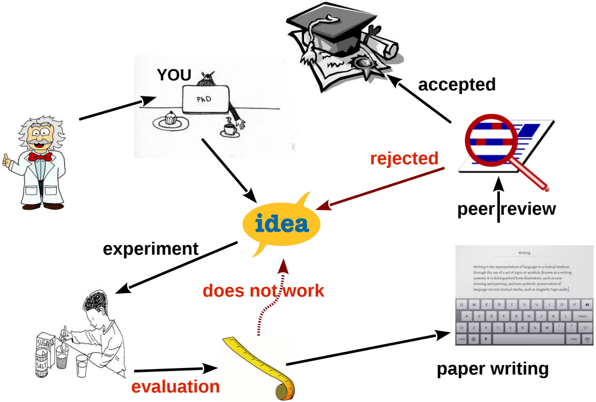
T(s, i) is the time to download them

3. Quality based scheduling
   - prioritize high quality pages

Outline

How the “research game” is played

- Loop for experiment-driven research
- Evaluation on a certain benchmark plays key role in the loop



Recall, precision and F-measure

- True Positive (TP): the number of relevant documents retrieved
- False Negative (FN): the number of relevant documents missed
- False Positive (FP): the number of irrelevant documents retrieved
- True Negative (TN): the number of irrelevant documents not retrieved
- Given the documents we consider (top-K), and relevant document R

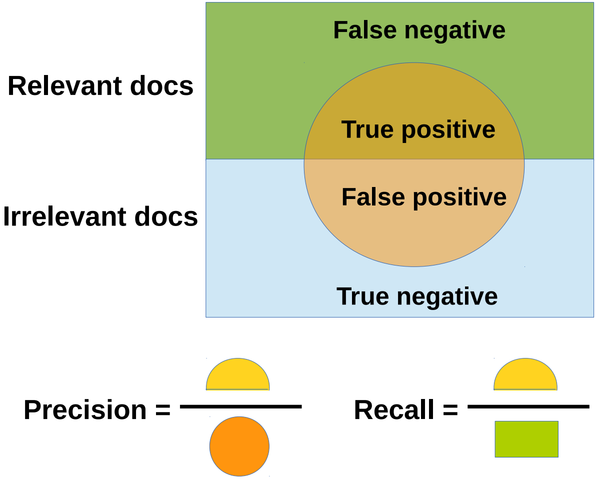
$$Recall = \frac{TP}{R} \tag{3}$$

$$Precision = \frac{TP}{K} \tag{4}$$

- F-measure is further defined as

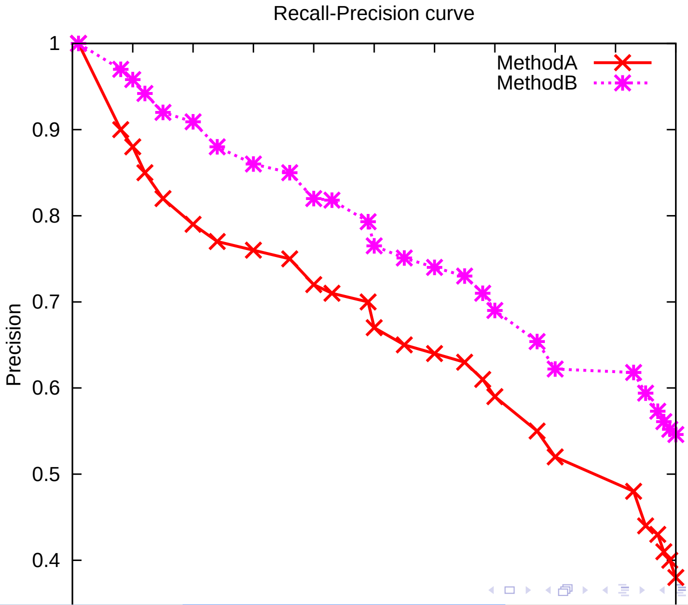
$$\text{F-measure} = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \tag{5}$$

Recall and precision illustration



- In classification task, the definition for ‘Precision’ changes

Curve of Recall V.S. precision



Average Precision

- Rankings of relevant docs are explicitly considered
- In practice, users are more sensitive to precision
- In-born advantage for a search engine:
  users have no knowledge about recall
- Average Precision is such a measure fits in
- Average Precision (AP) is defined as

$$AP(i) = \frac{\sum_1^i (1)}{i} \tag{6}$$

- mean Average Precision (mAP) is defined as

$$mAP = \frac{\sum_{i=1}^{K} AP(i)}{K} \tag{7}$$

Exercise

- Given total num. of relevant docs is 10

Top	Relevancy
1	1
2	0
3	1
4	0
5	0
6	0
7	1
8	1
9	0

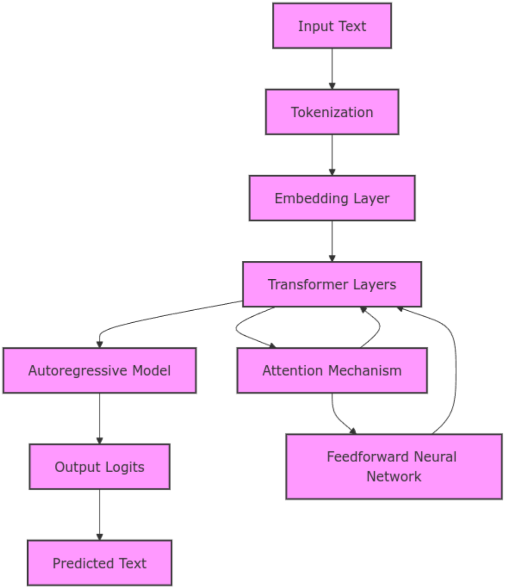
- See Recall=?, Precision=? and mAP=?

Outline

What is ChatGPT?

- ChatGPT is a conversational AI developed by OpenAI
- Based on the GPT (Generative Pre-trained Transformer) architecture
- Trained to understand and generate human-like text

Framework of ChatGPT



Key Features

- Natural Language Understanding
- Contextual Awareness
- Versatile Applications (eg, support, content creation)

How Does it Work?

- Utilizes deep learning techniques
  - Transformer and Reinforce learning
- Processes input text and generates responses

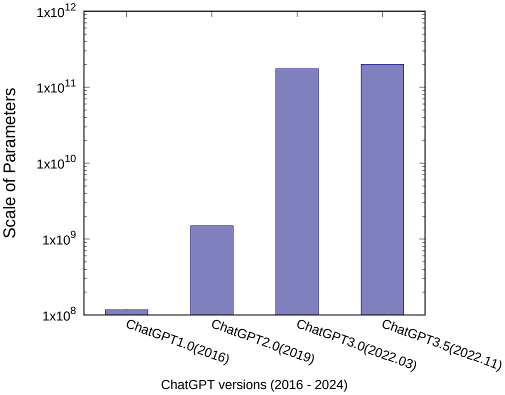
Jonh and his girl-friend are going to go to cinema to watch a movie.

- Learns from a diverse range of internet text
  1. Books
  2. Websites
  3. Wikipedia
  4. Research Papers
  5. Forums and Community Discussions

Applications

- Customer support automation
- Content generation and editing
- Personal assistants and chatbots

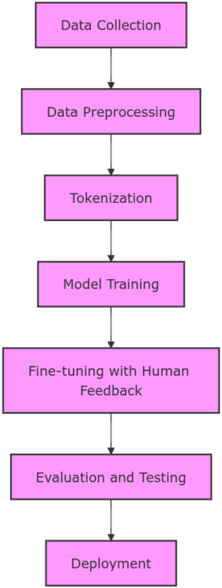
Challenges and Limitations



- May produce incorrect or nonsensical answers
- Sensitivity to input phrasing
- Ethical considerations and biases in AI
- The model of ChatGPT 3.5 takes at least 800G memory

Overview of GPT Training

- ChatGPT is based on the GPT architecture
- Trained using large corpora of text from the internet
- Focuses on predicting the next word in a sentence


Data Collection
Data Preprocessing
Tokenization
Model Training
Fine-tuning with Human Feedback
Evaluation and Testing
Deployment

Data Collection

- Utilizes diverse sources (books, articles, websites)
- Aims to cover a wide range of topics and language styles
- Ensures a broad understanding of human language

Preprocessing the Data

- Data is cleaned to remove low-quality content
- Tokenization: breaking text into manageable pieces (tokens)
- Transformation into a numerical format suitable for the model

Training Procedure

- Uses a technique called unsupervised learning
- Model is trained on predicting the next token based on context

  Jonh and his girl-friend are going to go to cinema to watch a <u>movie</u>.

- Backpropagation algorithm optimizes model weights

Fine-tuning

- Model further refined on specific datasets for accuracy
- Involves supervised learning with human feedback
- Enhances performance on conversational tasks and context

ChatGPT vs. Conventional IR

1. ChatGPT compress/encode the huge amount of knowledge into a model
2. Conventional IR index the information
3. Conventional IR index can be easily updated
4. Conventional IR index provide both the information and its source
5. Conventional IR index is cheaper

Outline

What is Retrieval-Augmented Generation?

- Most of the deep-models are not online model
- A hybrid approach combining retrieval and generation
- Enhances the capabilities of generative models
- Utilizes external knowledge sources to improve responses

How RAG Works

- Retrieves relevant documents based on user input
- Generates responses using both retrieved information and model knowledge
- Combines strengths of retrieval-based and generative methods

Key Components

- **Retrieval Component:** Searches for relevant documents
- **Generative Component:** Produces coherent and contextually relevant text
- **Integration Layer:** Merges information from both components

Benefits of RAG

- Access to up-to-date and specialized knowledge
- Improved accuracy and contextual relevance in responses
- Reduces the risk of generating incorrect information

Applications of RAG

- Question answering systems
- Conversational agents and chatbots
- Content generation and summarization tasks

Challenges and Considerations

- Dependence on the quality of retrieved documents
- Potential for biases in both retrieval and generation
- Need for efficient retrieval mechanisms

Q & A

Thanks for your attention!