Improving User Experiences for Wiki Systems

Master’s Defense
Advisor: Dr. Chris Pollett
Committee Members: Dr. Robert Chun, Dr. Philip Heller

Parth Patel | 12th December, 2022
Outline

• Purpose
• Background
• Preliminary work
• Recommendation System
• Yioop’s Recommendation System
• Enhanced Yioop’s Recommendation System
• Experiments
• Conclusion
• Future work
• References
Purpose

• Improve user experiences for Yioop’s wiki system

• Add emoji picker tool, UI testing and advertisement credits redeem features to Yioop

• Improve recommendation system in Yioop using Hash2Vec embedding

• Extend Yioop’s recommendation system to recommend wiki resources
Wiki systems are web applications that allow users to collaboratively manage information.

Popular wiki system - Wikipedia.

Yioop is an open source web application with features of search engine and wiki system.

Developed by Dr. Chris Pollett since 2009 using PHP.
Background - Groups and Threads in Yioop

• A group is a collection of threads and wiki pages.
• A thread contains a title and description.
• Access rules for a group govern users' activities.
Background - Wiki pages and Resource

- Wiki page contains textual data and files called as resources
- Different types of wiki pages
Preliminary work - Emoji Picker Tool

- Yioop provides direct messaging feature
- First version lacked support for emojis
- Implemented emoji picker tool
- Supports around 1500 emojis
- No external code dependency
- Improved experience for direct messaging
Preliminary work - UI Testing

- User interface affects the experience while interacting with an application.
- Yioop UI testing mechanism was outdated.
- A new UI testing project was created to ensure correctness of UI.
- Developed using Selenium and Node.js.
Yioop supports keywords based advertisement

Keyword advertisement bidding requires credits.

Credits are purchased using credit / debit cards.

Developed mechanism to convert unused credits back to real money.

Stripe services were used for transferring money to user bank account.
Recommendation System

• A recommendation system retrieves information based on user profile

• User profile is created based on information consumed by user

• Amazon, Netflix, Twitter, etc have recommendation system

• Greatly increases user experience

• Few types of recommender systems
  • Collaborative based
  • Content based
  • Hybrid
Yioop’s Recommendation System

- Yioop’s recommendation system recommends threads and groups
- Content based recommender system
- Recommendation media job runs at regular interval to update recommendations

Recommendations:

**Threads:**
- Trang_Phil%C3%A2n_Lo%E1%BA%A1lWiki Trang Tao Ral
- K%E1%BB%88m_tra_Cth%E1%BB%8B m%E1%BA%99l Trang Wiki Trang Tao Ral
- Chn_ph%C3%A2p_%C4%91%E1%BB%83 thu_th%E1%BA%A1lPh_th%C3%B4ng_tin_trang_Web_Wiki Trang Tao Ral

**Groups:**
- Group1
Yioop’s Recommendation System

• Leverages term frequency - inverse term frequency (TF - IDF) technique
• TF measure represent frequency of a term within a document
• IDF measure represent number of documents containing the term

\[
\begin{align*}
    tf(t, d) &= \log (1 + \text{freq}(t, d)) \\
    idf(t, D) &= \log \left( \frac{N}{\text{count} (d \in D : t \in d)} \right)
\end{align*}
\]

TF and IDF equation [12]
Yioop’s Recommendation System

- TF-IDF applied on BoW created from titles and description of threads
- TF-IDF scores calculated for each user based on their history
- Users view history captured in ITEM_IMPRESSION table
- Cosine similarity measured between users and threads TF-IDF
Enhanced Yioop’s Recommendation System

- Existing Yioop’s recommendation system required more time for processing data
- TF-IDF technique does not capture semantic meaning of words
- Recommendations generated were not accurate
- This project aims to enhance Yioop’s recommendation system
- Primarily Hash2Vec embedding technique will be used to replace TF-IDF
- Further extend recommendation functionality for wiki resources
Word Embeddings

• Word embedding is process of assigning vector to a word

• Simple embedding like CBoW or skip-grams are inefficient

• Advanced embedding techniques like Glove or Word2Vec utilize neural networks to compress CBoW embeddings to lower dimensions

• Neural networks require high end computing hardware, often taking longer time to process

• None of the above techniques can be used for Yioop
Hash2Vec Word Embedding

- Hash2Vec technique leverages hashing operation to calculate embeddings that can preserve semantic meaning [8].
- Hashing operation converts a word into a numerical value.
- The space complexity is $O(nk)$, $n =$ number of unique words and $k =$ size of embedding vector.
- The space complexity and processing time makes Hash2Vec ideal for Yioop.
Hash2Vec Implementation for Yioop

- Example
  - this is an example text corpus
- \( k = 2 \)
- Parameters value used
  - \( n = 200 \)
  - \( k = 5 \)
  - \( h = \text{md5}, \, \xi = \text{crc32} \)
  - \( f = (e^{-x})^2, \, x = d/\sigma \)

**Algorithm 1 Hash2Vec**

Parameters: \( n \) the embedding size, \( k \) the context size, \( h \) hash function, \( \xi \) hash sign-function and \( f \) aging function.

1: words \( \leftarrow \) Dictionary()
2: for every word \( w \) in text do
3:    if \( w \) \( \notin \) keys(words) then words[\( w \)] \( \leftarrow \) Array(\( n \))
4: for every context word \( cw \) with distance \( d \) do
5:    weight \( \leftarrow f(d) \)
6:    sign \( \leftarrow \xi(cw) \)
7: words[\( w \)][h(cw)] \( \leftarrow \) words[\( w \)][h(cw)] + sign \( \times \) weight

Hash2Vec algorithm [8]
Hash2Vec Implement for Yioop

• Preprocessing of terms in title and description of threads is done
• Applying Hash2Vec technique produced vectors for each unique term
• Threads vector are calculated using vectors of terms in its title and description
  • \( V(T) = V(T_1) + V(T_2) + \ldots + V(T_N) \)
  • Threads vectors are normalized
  • \( V(T) = \frac{V(T)}{|V(T)|} \)
  • \(|V(T)| = \sqrt{V(T) \times V(T)}\)

<table>
<thead>
<tr>
<th>ID</th>
<th>ITEM_TYPE</th>
<th>VECTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>3367</td>
<td>2</td>
<td>v2IKBM+EqtY/TXvOTRmxEd8y1yvGj/CLvKuHkq4JXeC/...</td>
</tr>
<tr>
<td>3368</td>
<td>2</td>
<td>v2M254152i/TX83ji+6/T8y2VnMPWj/AryKJd3cuNXa/...</td>
</tr>
<tr>
<td>3369</td>
<td>2</td>
<td>v2NU2GRGcw/TX/Hmc0WKZj8y2bW/sqmHykJ0eTw0k6j/...</td>
</tr>
<tr>
<td>3370</td>
<td>2</td>
<td>v2IIoTolQv4/TX+LMKbukD8y2Y891XOvyK/JtAZG73G/...</td>
</tr>
<tr>
<td>3372</td>
<td>2</td>
<td>v2VLPQzc0j/TX/5nFWIEz8y2dXNK4lvvyKJ8WQ+dRSj/...</td>
</tr>
<tr>
<td>3373</td>
<td>2</td>
<td>v2NKA3Emd8/TX+T7ukH7y2ZR7s99XyyKJsS3tsZK/...</td>
</tr>
<tr>
<td>3374</td>
<td>2</td>
<td>v2Tege2cw/TX/BBuFi.n2b8y2b6y4f8vyKJzda4J1HS/...</td>
</tr>
<tr>
<td>3375</td>
<td>2</td>
<td>v2IMZmaGc/TX7Z0UmnE8y2R3lylqwvvyKJPf6YIDC/...</td>
</tr>
</tbody>
</table>
Hash2Vec Implement for Yioop

• Different from existing system, user profile embedding vectors are calculated
• Embeddings of threads viewed by user are added and normalized
• Cosine similarities between user profile embeddings and threads embeddings are calculated
• Groups recommendation are calculated using embeddings of threads in it
Recommendation for Wiki Resources

• Recommendation functionality extended to recommend wiki resources

• A resource has a name and an optional description

• Apply Hash2Vec technique on the description of resources

• How to get descriptions for resources?
Yioop provides functionality to configure search sources.

Search sources are used by media jobs to fetch information from web.

Introduced Description search source to fetch description for resources.

Use XPath to fetch relevant information on given website.
Description Update Media Job

- Developed DescriptionUpdate media job to process description search sources
- Implemented an efficient mechanism to track resources without description
- Able to fetch descriptions using just the name of resources
- Selects the result with highest title match score to fetch required details
Extended Yioop’s Recommendation System

- Extended recommendation media job to apply Hash2Vec on fetched description of resources
- Similar mechanism leveraged to keep track of resources frequently viewed by users and have descriptions
- Resources with top 3 scores are recommended on home page
Experiments - Description Update Job

- Experiments conducted on 2 collections of resources
- First collection had total of 137 movies, TV shows and books released in 2022
- Second collection had 78 TV shows dating as back as 1950
- Configured 2 description search sources for IMDb and Goodreads website

<table>
<thead>
<tr>
<th>Collection</th>
<th>Correct</th>
<th>Wrong</th>
<th>Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection 1</td>
<td>130</td>
<td>None</td>
<td>43</td>
</tr>
<tr>
<td>Collection 2</td>
<td>45</td>
<td>33</td>
<td>None</td>
</tr>
</tbody>
</table>
Experiments - Resource recommendation

- Experiment was done on 137 resources in collection 1
- 5 test users were created
- 4 users interacted with 15 resources each and fifth user did not interact with any resource
- A profile label was attached to a user based on the resources interacted
- For example user interacting with only action movies is given Action profile label
### Experiments - Resource recommendation

<table>
<thead>
<tr>
<th>User</th>
<th>Profile</th>
<th>Relevant Recommendations</th>
<th>Irrelevant Recommendations</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Action</td>
<td>2</td>
<td>1</td>
<td>0.67</td>
</tr>
<tr>
<td>2</td>
<td>Science, Fiction, History and Romance</td>
<td>1</td>
<td>2</td>
<td>0.33</td>
</tr>
<tr>
<td>3</td>
<td>Drama</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Thriller, Comedy, Family, Anime</td>
<td>1</td>
<td>2</td>
<td>0.33</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>7</td>
<td>5</td>
<td>0.58</td>
</tr>
</tbody>
</table>
Experiments - Threads and Groups Recommendation

- Experiments were conducted to compare accuracy of enhanced and existing recommendation system
- Two instances of Yioop server installed on same machine
- 5 groups with 10 threads each were created on both instances
- Thread contained descriptions on 5 categories
  - Software development
  - Political affairs
  - Sports
  - Stock market
  - Gaming
## Experiments - Threads and Groups Recommendation

<table>
<thead>
<tr>
<th>Group</th>
<th>Software Development</th>
<th>Gaming</th>
<th>Political affairs</th>
<th>Stock market</th>
<th>Sports</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>Mixed</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>Gaming / Sports</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>Political / Stock market</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Software development</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>Mixed</td>
</tr>
</tbody>
</table>
Experiments - Threads and Groups Recommendation

- 5 test users were created on both instances
- User was member of same group and viewed same 5 threads
- Recommendation job completed processing in 8 seconds on instance with enhanced recommendation system
- Recommendation job took 24 seconds to finish processing on another instance with existing recommendation system
- Results summarized on next slide
### Experiments - Threads and Groups Recommendation

<table>
<thead>
<tr>
<th>User</th>
<th>Enhanced Recommendation System Accuracy - Thread</th>
<th>Existing Recommendation System Accuracy - Thread</th>
<th>Enhanced Recommendation System Accuracy - Group</th>
<th>Existing Recommendation System Accuracy - Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.67</td>
<td>0.33</td>
<td>1</td>
<td>0.33</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.67</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0.33</td>
<td>0</td>
<td>0.67</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0.67</td>
<td>0.67</td>
<td>1</td>
<td>0.67</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0.67</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Average</td>
<td>0.73</td>
<td>0.47</td>
<td>0.93</td>
<td>0.6</td>
</tr>
</tbody>
</table>
Conclusion

- Implemented new features which contributed to improving user experiences for Yioop
- Hash2Vec embedding technique was studied and implemented for Yioop recommendation system
- Enhanced recommendation system using Hash2Vec performs better than existing recommendation system
- Extended recommendation system for wiki resources using novel approach of extracting descriptions from web
- Processing time for recommendation job is decreased
Future work

• Multiple areas of future work building upon this project

• Description update job can be improved
  • Compare results from multiple sources to select best match
  • Incorporate other information of resource to improve matching process

• A weight function can be designed to weight term embeddings while calculating threads and resource embeddings

• Introduce mechanism to take feedback from user for recommendations
References


[3]. M. Anirudh, “An open source direct messaging and enhanced recommendation system for Yioop”, San José State University, Dec 2021


Thank You!