Book Recommendation and Summary System

Shutong Jiao
Xuezhang Wu
Lanchun He
Yu Lei

ABSTRACT

In this project our team builds a book recommendation system based on Collaborative Filtering, Content Based Filtering and Automatic Text Summarization. We used Goodreads Dataset from UCSD Book Graph. First we implemented advanced hybrid recommendation system by using Content Based Filtering to solve Cold Start problem and increase the diversity of recommendation. Then we extracted summaries and keywords from book reviews using TF-IDF and BM25 based Tag method,Latent factor models and TextRank Algorithm.

KEYWORDS

recommendation system, text summarization, collaborative filtering, content based filtering, textrank

ACM Reference Format:
Shutong Jiao, Xuezhang Wu, Lanchun He, and Yu Lei. 2019. Book Recommendation and Summary System. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Recommendation system is not a new idea and it provides benefits for both seller and customers. Specifically, book recommendation system can be useful not only for book sellers but also for libraries and schools. The huge number of books available on the internet to some extent bring troubles to readers as readers are harder to find the right book to read. A good book recommendation system could help reader in this way and therefore improving the life of a reader. In this project, a book recommendation system is put forward along with a summarization system that provides user extracted summaries from book review.

2 RELATED WORK

First of all, an easy way to make book recommendation is to recommend book as common item using generic recommendation example. For example, Amazon employs the same recommendation technique to work with all their products[3]. Other specific recommendation use data like library loans[7], stylometry feature[6] and social media[?]. However, a recommendation without reason is not sufficient. Readers are curious about the reason behind the recommendation. A way to provide the reason is to let users have a quick understanding of the content of the book. The official description could help but clearly this is a biased description because it is written by the author and the publisher. Another way to do this is to automatically generating summary form user review of this book using NLP techniques. As the summary comes from user's review, they will be helpful for the reader to decide if this book is interesting to him.

3 DATASET,GITHUB REPO AND POSTER

A link of the GoodReads dataset could be found here. We only used the poetry dataset. The GitHub repo is here and the poster is here.

4 RECOMMENDATION SYSTEM WORKFLOW

4.1 Input User

The input user is an existing user in the dataset which has rating records for books

4.2 Collaborative Filtering/ Content Based Filtering

If user don't have enough rated book record, we combine CF and CB to generate recommendations

4.3 Similarity Based Deduplication

To avoid boring users by recommending books which have similar plots. We hope use CB based filter to screen out books which has similar description.

4.4 Automatic Text Summarization

Generate summary from user's reviews of the recommended books

4.5 Recommended Book

The outcome is the recommended book with book id, book title, and summary for the user

5 COLLABORATIVE FILTERING AND MATRIX FACTORIZATION

The naÃŕve recommendation system provides personalized recommendation based on user's historical ratings by Collaborative Filtering. It calculates the ratings of un-rated books from known ratings, then selects books that have high predicted ratings to recommend.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Method	Cosine Similarity	Pearson Correlation	Pearson Correlation (scaled)	Matrix Factorization
MAE	0.6646	0.6928	0.6562	0.6203
RMSE	0.8687	0.9350	0.8625	0.8136

Table 1: Overall evaluation of collaborative filtering methods

5.1 Methods

5.1.1 User-Based Collaborative Filtering. User-Based Collaborative Filtering uses the logic that people with similar ratings on books will have similar taste. It creates the artificial ratings by finding similar users to that active user (to whom we are trying to recommend a book). The features we used to evaluate this similarity between two users are cosine similarity, Pearson correlation, and optimal Pearson correlation which is scaled by the number of co-rated books.

5.1.2 Matrix Factorization âĂŞ Stochastic Gradient Descent. Matrix factorization is the algorithms work by decomposing the sparse user-item interaction matrix into the product of two lower dimensionality rectangular matrices. Then it uses those dense matrices to calculate those unknown ratings in the user-item matrix.

The algorithm that this system used to get the factorized matrices is called Stochastic Gradient Descent algorithm. When using Stochastic Gradient Descent for Collaborative Filtering, it wants to estimate two matrices âĂŞ the latent user feature matrix P and the latent book feature matrix Q. After having estimated P and Q, we can then predict the unknown ratings by taking the dot product of the latent features for users and books. For getting the estimated P and Q, we need use stochastic gradient descent to loop over every observation in the training set and update P and Q. it generates two random initial matrices. During each iteration, those matrices would be updated by the error (difference) between the actual ratings and the temporary predicted ratings. Comparing with Alternating least squares, stochastic gradient descent algorithm is easier and faster.

5.2 Overall Result and Evaluation

We can use Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to evaluate the performance of those algorithms used in our naÃŕve recommendation system. The dataset contains 36374 books and 282415 users. The result is shown in the table below. The matrix factorization method has the lowest MAE and RMSE, which means that it performs the best. And the user-based collaborative filtering with scaled Pearson correlation performs the second.

5.3 Example from actual user

This part uses two actual users to show the recommendation result. Figure 1 and Figure2 show the ratings of those 2 users. User 1 rated 7 books, and we can find that the books he concerns is historical and classic poetry. User 2 only rated 3 books, and all of them are related to inner emotions, such as romance and affections between families.

User1	Book ID	Book	Rating Score
1	764332	Jason and the Golden Fleece	4
2	1519	The Oresteia (Ορέστεια, #1-3)	4
3	1715	Metamorphoses	4
4	12914	The Aeneid	4
5	1371	The Iliad	5
6	1381	The Odyssey	5
7	2696	The Canterbury Tales	4

Figure 1: User 1's book ratings

User2	Book	Rating Score
1	Sad Girls	2
2	Memories	4
3	Love & Misadventure	3

Figure 2: User 2's book ratings

Then we set the books they didn't rate as the test set and using the basic recommendation system to find the recommendation books for those two users. The results are shown in Figure 3 and Figure 4. We can find that this system recommends historical and classic books for user 1, such as The Divine Comedy, Hamlet, and Paradise Lost. Most of the poetry recommended to user 2 are about inner emotions, such as Lullabies, The Universe of Us, and I Wrote This For You. The basic recommendations for those two actual users are personalized and perform well.

User 1's Recommandations

Rank	Book ID	Book Name
1	6656	The Divine Comedy
2	15645	Inferno (The Divine Comedy #1)
3	1420	Hamlet
4	23919	The Complete Stories and Poems
5	333706	The Odyssey
6	15997	Paradise Lost
7	80410	Four Quartets
8	42051	The Complete Sonnets and Poems
9	42038	Shakespeare's Sonnets
10	782580	The Complete Poetry and Prose
11	269322	The Raven and Other Poems
12	118389	The Love Song of J. Alfred Prufrock and Other Poems
13	75504	The Collected Poems
14	23913	The Marriage of Heaven and Hell
15	23912	The Complete Poems
16	881508	Pan Tadeusz
17	142080	Collected Poems, 1909-1962
18	5931	The Essential Neruda: Selected Poems
19	1463	Euripides V: Electra / The Phoenician Women / The Bacchae
20	222035	Poemcrazy: Freeing Your Life with Words
21	134018	The Complete Poetry and Selected Prose
22	22221	The Iliad
23	34080	The Waste Land
24	94578	The Gay Science
25	53022	The Collected Poems of W.B. Yeats
26	1182095	The Branch Will Not Break
27	119234	Lord Byron: The Major Works
28	139004	Essays and Poems
29	19154	The Doré Illustrations for Dante's Divine Comedy
30	111044	Stop Pretending: What Happened When My Big Sister Went Crazy

Figure 3: Recommend books for User 1

User 2's Recommandations

Rank	Book ID	Book Name
1	22151696	Lullabies
2	29431081	The Universe of Us
3	23513349	Milk and Honey
4	25384844	Black Butterfly
5	23434371	Beautiful Chaos
6	13123245	B
7	13105527	I Wrote This For You
8	35606560	The Sun and Her Flowers
9	25746714	The Type
10	19230408	I Wrote This For You: Just the Words
11	23534	Love Is a Dog from Hell
12	18288210	No Matter the Wreckage
13	29457318	Habang Wala Pa Sila: Mga Tula ng Pag-ibig
14	13376363	Teaching My Mother How to Give Birth
15	32468495	Pillow Thoughts
16	7824768	ليتها تقرأ
17	6017893	قهوة وشيكولاتة
18	980426	Love Poems
19	20821097	Chasers of the Light: Poems from the Typewriter Series
20	11625	Ariel: The Restored Edition
21	23522212	Mouthful of Forevers
22	29758714	Dirty Pretty Things
23	31443393	Note to Self
24	6944946	يوميات امرأة لا مبالية
25	25986828	Today Means Amen
26	1294049	Love Songs
27	24688932	All The Things I Never Said
28	25334576	Grief is the Thing with Feathers
29	19265831	Hello, Baby
30	26850255	To The Women I Once Loved

Figure 4: Recommend books for User 2

However, this system also has some problems. The first one is the system recommend some books that is same or very similar to books in this userâĂŹs rating list. Their content is very similar so that we donâĂŹt want to recommend it but the book id of them is different so that it wonâĂŹt be screen out. For example, we recommend The Odyssey and The Iliad for user 1, but he rated those two books (maybe different edition) before. The second problem is that this system is only based on the userâĂŞitem interaction data, so that the recommendation will also concerns about the ratings, but it may ignore some inner relations such as content. User 1âĂŹs recommendations are about historical and classic poetries, but it doesnâĂŹt have a strong trend to Greek epic poetry which is shown in his rating history. This weakness can be solved by the following part âĂŞ content based filtering.

6 CONTENT BASED FILTERING

Typical Recommendation system normally must solve two problems: cold start and lacking diversity in recommendation. To solve these two problems, we combined Collaborative Filtering and Content Based filtering together.

TF-IDF, Word2Vec and Sentence2Vec: three methods are used to produce matrix representation for description of each book.

6.1 Methods

6.1.1 TF-IDF. The first method we tried is TF-IDF with Cosine Similarity. We use description of each poetry to calculate the TF-IDF score for each word in each document. These scores are the structured item representation. The vector space model âĂŞ cosine similarity is then applied to get the similarity between every poetry. The result is not the most ideal one since it only consider words appeared in each document.

6.1.2 Word2Vec. The second method we tried is Word2Vec. We used Word2Vec model to average the vectors of all the words in a sentence. Each sentence is tokenized to words, vectors for these words can be found using glove embeddings and then take the average of all these vectors. This technique has performed decently, but this is not a very accurate approach as it does not take care of the order of words.

6.1.3 Sentence2Vec. The last method we tried is Sentence2Vec - Infersent, which is a sentence embeddings method that provides semantic sentence representations. It is trained on natural language inference data and generalizes well to many different tasks.

6.2 Results

We create a vocabulary from the training data and use this vocabulary to train Infersent model[2]. Once the model is trained, it provides sentence as input to the encoder function which will return a 4096-dimensional vector irrespective of the number of words in the sentence. Compared to Word2Vec which weighs words evenly, Sentence2Vec weights words differently so each word has its own "importance". The figure below shows the visualization of words importance of sentence: "Barack-Obama is the former president of the United-States."

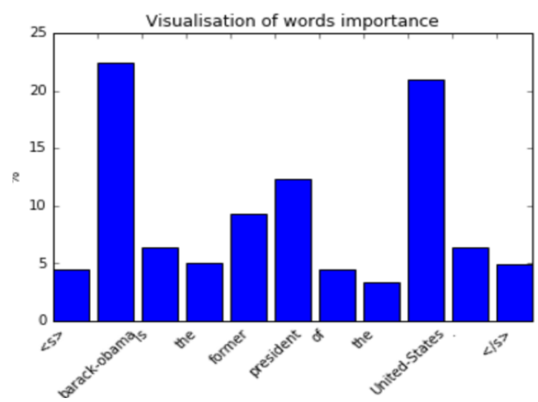

Figure 5: Visualization of words importance of sentence from Sentence2Vec model

By giving a target book, we calculated books’ cosine similarity with this book using TF-IDF, Word2Vec and Sentence2Vec.The book which has the highest cosine similarity using these three methods is shown below.

Target book: Season Songs is a book about collection of poems described four seasons. Using Word2Vec method, it recommended us a book about collection of five previous books. Using TF-IDF method, it recommended a book only about collection poems. And only when using Sentence2Vec method, we can get a book about poems described four seasons. Therefore, we say Sentence2Vec has the best performance among 3 methods, and we applied this method into our system. Figure below shows the Content Based Filtering recommendation result for book "Memories".

Target Book:	1293847, Season Songs
Description	collection of twenty-eight poems grouped to represent the four seasons
Word2Vec Rec	Hinge & Sign: Poems
Description	words and forms challenge each other in a collection of new poems and selections from the author's five previous books
TF-IDF Rec	The Collected Poems of Sara Teasdale
Description	collection poems
Sentence2Vec Rec	Haiku Reflections: The Four Seasons
Description	a collection of 120 haiku poems, exploring and celebrating the wonder and unique beauty of each of the four seasons.

Table 2: Target book and its most similar books using different method

Rank	Book ID	Book Title
1	303865	Collected Poems of Emily Dickinson
2	1291581	The Journals of Susanna Moodie
3	960036	The Lady of Shalott
4	644996	The Legend of Mulan: A Folding Book of the Ancient Poem That Inspired the Disney
5	1370171	When We Were Very Young
6	20801053	When We Were Very Young
7	3406085	When We Were Very Young
8	44450	The Portable Dorothy Parker
9	170372	Love Poems from the Japanese
10	160889	Illustrated Treasury of Read-Aloud Poems for Young People
11	33626006	Selected Poems
12	133907	Poetry for Young People: Langston Hughes
13	509454	Climb Into My Lap First Poems to Read Together: First Poems to Read Together
14	7083492	The Canterbury Tales (Barnes & Noble Classics Series)
15	13542892	Madame X

Figure 6: Content based Filtering Recommendation for book Memories

Figure below shows the result of combination of Collaborative Filtering and Content Based Filtering. In this figure, we weighted Collaborative Filtering and Content Based Filtering both for 0.5. In the future, we plan to implement a dynamically weighting system based on number of books that user has rated. If user has rated only a few books, we will increase the weight of Content Base filtering, and if the user has rated many books, the weight for Collaborative Filtering will be increased.

Rank	Book ID	Book Title
1	303865	Collected Poems of Emily Dickinson
2	22151696	Lullabies
3	1291581	The Journals of Susanna Moodie
4	29431081	The Universe of Us
5	960036	The Lady of Shalott
6	23513349	Milk and Honey
7	644996	The Legend of Mulan: A Folding Book of the Ancient Poem That Inspired the Disney
8	25384844	Black Butterfly
9	1370171	When We Were Very Young
10	23434371	Beautiful Chaos
11	20801053	When We Were Very Young
12	13123245	B
13	3406085	When We Were Very Young
14	13105527	I Wrote This For You
15	44450	The Portable Dorothy Parker

Figure 7: Recommendation list after combing Content based Filtering and Collaborative Filtering

After the Combination, we calculate the cosine similarity for each two books in the recommendation list iteratively. If the similarity of two books is too high, then we screen out the book which has a lower ranking. We hope that by using this algorithm, we can screen out books which has similar description and hence increase the diversity of recommendation list. Figure X shows the final result after screening

Rank	Book ID	Book Title
1	303865	Collected Poems of Emily Dickinson
2	22151696	Lullabies
3	1291581	The Journals of Susanna Moodie
4	29431081	The Universe of Us
5	960036	The Lady of Shalott
6	23513349	Milk and Honey
7	644996	The Legend of Mulan: A Folding Book of the Ancient Poem That Inspired the Disney
8	25384844	Black Butterfly
9	1370171	When We Were Very Young
10	23434371	Beautiful Chaos
11	13123245	B
12	13105527	I Wrote This For You
13	44450	The Portable Dorothy Parker
14	35606560	The Sun and Her Flowers
15	170372	Love Poems from the Japanese

Figure 8: Final filtering to screen out books which has similar plot using Sentence2Vec

7 AUTOMATIC TEXT SUMMARIZATION

7.1 Motivation

Book review provides a rather unbiased insight into books. When we recommend books to users, users are curious about the reason behind recommendation. Text summarization on book reviews provide such a reason why this specific book is recommended. On the other hand, manual summarization is laborious but automatic summarization has fewer biases, is faster, more scalable and more cost-efficient.

7.2 Related work

For text summarization, the basic approaches can de divided into extractive ones and abstractive ones as shown in Figure 10. For extractive ones, the goal is to select most important sentences from the document and rearrange the summary using the extracted sentences. For abstractive ones, the goal is to learn internal language representation along with paraphrasing to create a more human-like summary. However, the advantage of extractive method is that it has high interpret ability and it is also mostly unsupervised. As a result, extractive approaches are explored and showed satisfying result.

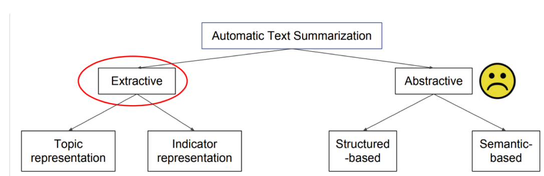

Figure 9: Diagram of text summarization

7.3 Approaches

Here, the example of the summary and keywords comes from a review to The Divine Comedy by Dante. The Divine Comedy describes Dante's descent into Hell with Virgil as a guide; his ascent of Mount Purgatory and encounter with his dead love, Beatrice; and finally, his arrival in Heaven.

7.3.1 TF-IDF and BM25 based Tag method. By intuition, importance sentences usually contain many tag words or important words. Then the task is to find tag words. Here tag words are defined as words that have high TF-IDF / BM25 value. Noteworthy, in TF-IDF, the term frequency is the frequency of word in this review and the document frequency is the frequency of the word in the whole English corpus. Then we select top tag words and choose sentences that have most tag words, normalized by another TF-IDF count. This method as you can see from table below is naive but is easy to improve with parameter tuning and serves to be a good baseline.

Summary	The opposite of a great truth is another truth." - Niels BohrI was thinking about Dante the other day and wondering how one could approach him from the angle of a GoodReads review.I dowonder what he would have thought if he had been able to learn that many leading religious figures, even in the early 21st century, reject a large partof science as being somehow unreligious.
Keywords	'dante', 'dawkins','heaven', 'universe', 'would', 'religious', 'one', 'wondering', 'could',

Table 3: summary and keyword from TF-IDF and BM25 based Tag method

7.3.2 Latent factor models. Firstly, we construct a $m \times n$ sparse matrix whose row corresponds to terms and column corresponds to sentences. With matrix factorization, we could decompose this matrix to a $m \times r$ matrix and a $r \times n$ matrix. Here the row in the $r \times n$ matrix corresponds to "concepts" and the column corresponds to sentences. Therefore we could select sentences with highest sum of "concepts" and generate the summary. Similarly using the $m \times r$ matrix we could also generate the keywords. The latent factor model is shown in Figure 11 and the result is shown in Table 4.

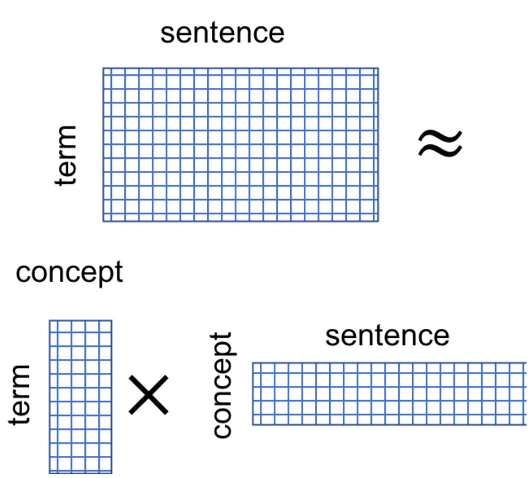

Figure 10: Latent factor models

Summary	The opposite of a great truth is another truth." "Now of course, I am aware that Dante was deeply immersed in the Christianworld-view, and Dawkins is famous for being the world's most outspoken atheist.Dante was a Christian to the core of his being, but he was furious with the way the Church was being run; he put several of its leaders, notably Pope Boniface VIII, in Hell
Keywords	'dante',,'could', 'wondering', 'would', 'heaven', 'one', 'let', 'light',

Table 4: summary and keyword from Latent factor models

7.3.3 TextRank Algorithm. TextRank is an unsupervised algorithm based on weighted-graphs built on top of the popular PageRank algorithm[4]. The basic steps are :

- Pre-process the text: remove stop words and stem the remaining words.
- Create a graph where vertices are sentences
- onnect every sentence to every other sentence by an edge. The weight of the edge is how similar the two sentences are.
- Run the PageRank algorithm on the graph
- Pick the vertices(sentences) with the highest PageRank score

And it has also been shown that BM25 as a similarity measure between sentences achieves the best performance[1]. The resulting summary and keywrods is shown in Table 5.

Summary	The opposite of a great truth is another truth." - Niels Bohr. I was thinking about Dante the other day and wondering how one could approach him from the angle of a GoodReads review. For Dante, there didn't seem to be any opposition between religious faith and science - they were part of the same thing.
Keywords	religious day days world truth wondering wonder wonderful background geocentric fluctuations

Table 5: summary and keyword from TextRank with BM25

In fact, the original textrank algorithm can be improved in a number of ways like using lemmatization instead of stemming, incorporating Part-Of-Speech tagging and Named Entity Resolution[5]. The improved algorithm, also called PyTextRank, works as the step shown below:

- Perform Part-of-Speech Tagging and lemmatization for every sentence in the document.
- Construct a graph where nouns, adjectives, verbs are vertices. The weight of each link is given by skip-grams or repeated instances of the same root
- Use PageRank to noun phrases known as feature vector which have high probability of inbound references
- Calculates a score for each sentence by jaccard distance between the sentence and feature vector

- Pick the sentences with the highest score

The figure below shows the graph generated in step 2 and the summary and keyword is shown in Table 6.

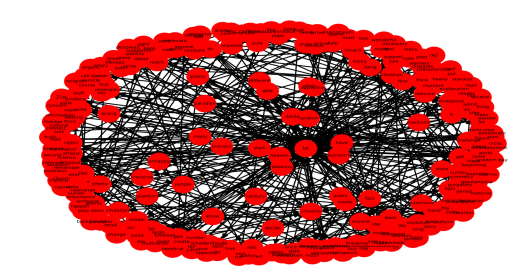

Figure 11: Diagram of text summarization

Summary	The opposite of a great truth is another truth.", But it's not quite as crazy as it first may seem. Obviously we would n't havethe old geocentric model of the Universe- it would be bang up to date. I think there is now far more material for an ambitious poet to work with than there was in the 14th century. Finally, we reach the Heaven of the Multiverse, andfind that we are just one of many different universes.
Keywords	ambitious poet, most outspoken atheist, actual fact, more material, heaven, universe,life

Table 6: summary and keyword from improved TextRank

7.4 Evaluations

In order to evaluate the performance of these summary method using metrics like Rouge-N, a gold summary is needed. However, we do not have gold summaries. Alternatively, we could assess the performance by looking at the keywords extracted from the review.

For example, TF-IDF based tag method and latent factor models give keywords like 'dante', 'dawkins', 'heaven', 'universe', 'would'. We could see that some of them are relevant like 'dante' and 'heaven' and some are not relevant like 'would'. Textrank method gives keywords like 'religious' 'world' 'truth' 'wondering' 'wonder' 'wonderful' 'background' 'geocentric' 'fluctuations', which are more relevant. However, some words like 'wondering' and 'wonder' comes from same lemma. In the improved textrank method, lemmatization solved this problem and also with POS tagging it could extract key phrases rather than single keyword. The key phrases given from this method are 'ambitious poet', 'most outspoken atheist', 'actual','fact', 'more' 'material', 'heaven', 'universe','life'. We could see that these key phrases are most related to the review and the book by intuition and therefore we believe that improved textrank method gives the best performance.

8 CONCLUSION

8.1 Challenges

The first challenge here is how to process huge input data. The collaborative filtering method and the sentence2vec model both require a lot of computation. The collaborative filtering method also require a lot of space.We have tried using AWS and Google colab and the result is also not good. At last we decide to particularly choose poetry from Coloreds dataset because this is the smallest dataset. Even so, it has 282115 users and 36374 books. Another challenge is about evaluation for the summarization part. As there is no gold standard summary for book review, metrics like ROGUE-N is not applicable here. What we do here is to provide keywords and compare the keywords with the theme of the book and the review. While this method do not provide a numerical result, it is sufficient enough to compare each summarization method.

8.2 Future Work and Open Issue

Firstly, in order to have better scalability,reliability, availability and solve the first challenge, a distributed recommendation system is a necessity. Depending on the scenario of application, this distributed recommendation may have different architecture and this is an open issue worth more discussion.

Besides, there also exist many future work for text summarization system. For unsupervised approach, we could use heuristic approach like associating sentences a score based on features like sentence position, sentence length and title feature. For supervise approach, with sufficient data, we could build a neural network with these features to predict sentences extracted.

REFERENCES

[1] Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. 2016. Variations of the Similarity Function of TextRank for Automated Summarization. *CoRR* abs/1602.03606 (2016). arXiv:1602.03606 http://arxiv.org/abs/1602.03606

[2] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364* (2017).

[3] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 1 (2003), 76–80.

[4] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing.*

[5] Paco Nathan. 2016. PyTextRank, a Python implementation of TextRank for text document NLP parsing and summarization. https://github.com/ceteri/pytextrank/.

[6] Maria Soledad Pera and Yiu Kai Ng. 2014. How can we help our K-12 teachers?: using a recommender to make personalized book suggestions. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 02.* IEEE Computer Society, 335–342.

[7] Keita Tsuji, Nobuya Takizawa, Sho Sato, Ui Ikeuchi, Atsushi Ikeuchi, Fuyuki Yoshikane, and Hiroshi Itsumura. 2014. Book recommendation based on library loan records and bibliographic information. *Procedia-social and behavioral sciences* 147 (2014), 478–486.