Combining natural language processing and topic modeling techniques for filtering related news

Kai Sum Yau
Faculty of Science
Queensland University of Technology
2, George Street, Brisbane, 4001, QLD, Australia

***Abstract*—A large volume of text content is available over the world wide web. Social media, digitized libraries and news portals contain an enormous amount of text contents, and it is challenging to filter the required contents based on the user's interest. In traditional information filtering concept, it is assuming one interest one topic, it is not real. Therefore, this paper proposes a novel mashup model which is combination of natural language process and latent dirichlet allocation to extract user's interest from the user's reading history. Assuming one user has multiple interest and topics.**

***Keywords*—information fitlering; information retrieval; topic modelling; LDA; Natural Language Processing; text mining; word embedding**

I. INTRODUCTION

Nowadays, people always upload and share their paper in the Internet, such as news and articles. A large volume of textual data is available over the world wide web, then, a strong filtering is convenient for us. Therefore, information filtering (IF), topic modeling and natural language processing (NLP) are important techniques used to develop the strongest filter. IF is a system to manage large information flows including removing redundant or unwanted information from the textual data stream [1-2]. Topic modeling is an unsupervised machine learning technique by extracting the words from the document and finding the topics to represent the document. NLP is a technique for supporting topic modeling model. According to Rania Albalawi et al. [3], "NLP is a field that combines the power of computational linguistics, computer science, and artificial intelligence to enable machines to understand, analyze, and generate the meaning of natural human speech". It means NLP is trying to use some formula to calculate the relationship between the words in a document.

IF, topic modeling and NLP are growing fast, many different methods and algorithms are proposed, such as the famous model is Latent Dirichlet Allocation [4]. The challenges are the handling of disambiguation, noise text and the length of content affecting the effective of the model. Therefore, I design to combine these two categories thing together information filtering model.

The aims of the research are finding the best combination of NLP and LDA and the best parameters of the model, and evaluating the performance of the model. A novel idea is proposed to combine NLP technique and LDA model to extract the user's interest from the user's reading history and then filter the relevant news in another dataset which contains the news that is the user never reads. Therefore, the research problem is the correctness of filtering the news based on the user's reading history. There are two research questions: "Which combinations of the existing information filtering and topic modeling approaches can be suitable for filtering related news stories based on user's reading history?" and "How does the novel mixed approach improve extracting user's interest?".

Since there are many existing libraries and algorithms to implement the NLP and topic modeling techniques, the existing libraries and model is proposed to use in the research. In the process of the experiment, two different NLP techniques are tested which are count vectorizer and tf-idf vectorizer. The experimental result is count vectorizer cross LDA model has a higher accuracy and lower perplexity. It means it is better than tf-idf vectorizer.

This paper makes the following contributions:

- I review scholarly articles related to IF, topic modeling and NLP from 2003 to 2021.
- I explore the mashup approaches of NLP and topic modeling methods that are commonly used in information filtering and text mining, called LDA, count vectorizer and tf-idf vectorizer.
- I evaluate two combination methods, count vectorizer cross LDA and tf-idf vectorizer cross LDA. The standard of evaluation is the accuracy of score of documents, and the perplexity of the model.
- The aim is comparing and evaluating the mashup methods to find the effective approach in extracting interest from the user to filter the relevant news.

This paper is organized as follows. Section II describes the related works and the review of literatures. In section III, the novel mashup model is introduced. The results and discussion of the experiment is shown in section IV. The last section concludes the research methodology, suggestions for improvements, the limitations of the novel model and the outline of future work.

II. RELATED WORK

To understand the process and challenges of IF, NLP and topic modeling, I reviewed the literatures of these area. In traditional IF models, there are term-based models, such as

BM25 [8], SVM and Rocchio, and pattern-based models, such as closed pattern and sequential pattern. However, traditional IF models assume the user's interest is only related to one topic [1]. It does not make sense. In this paper, I propose extracting multiple topics from the user to represent multiple interests. Therefore, I reviewed the area of topic modeling and NLP.

In topic modeling, there are many studies discussing different algorithms and techniques including probabilistic, semantic and pattern-based techniques.

LDA is the most popular technique in probabilistic models, and many topic models are based on LDA to implement an enhancement LDA model. For example, Tang et al. [7] introduced a novel structure called four-layer hierarchical Bayesian structure, and two novel supervised topic models called Conceptualization Latent Dirichlet Allocation (CLDA) and Conceptualization Labeled Latent Dirichlet Allocation (CLLDA). The four-layer hierarchical Bayesian structure is based on the basic assumption of LDA which is three-layer hierarchical Bayesian structure, and adding a concept layer between topics layer and words layer. CLDA and CLLDA are based on the four-layer hierarchical Bayesian structure to improve LDA and Labeled LDA (LLDA) models.

In semantic techniques, Geeganage et al. [6] proposed a Semantic-based Topic Representation using Frequent Semantic Patterns (STRuFSP) to represent the semantic of the topics. STRuFSP is used to generate a semantic-based topic representation for a textual dataset. This is the latest paper when I was writing this paper, so it is the state-of-the-art method. The article also compares the perplexity and coherence between LDA, CLDA [7], Probase-LDA and Maximum Matched Pattern-based Topic Model (MPBTM) [1]. The conclusion of this paper is handling the unavailable words in Probase is important in semantic topic modeling, and handling disambiguation is a challenge in the future.

In pattern-based techniques, Gao et al. [1] presented a novel information filtering model called MPBTM. MPBTM has four main features which are user information must include terms of multiple topics, all topics are represented by patterns, topic models produce the patterns by organizing the statistical and taxonomic features, the patterns of this model are the most discriminative and representative. Comparing to the state-of-the-art models, MPBTM provides a better document modelling and relevance ranking.

In 2017, Malek Hajjem and Chiraz Latiri [9] proposed to combine Information retrieval (IR) approach and LDA for aggregating tweets in order, which improves the quality of LDA-based topic modeling. However, the experiment target is short text.

In 2011, Doumit and Minai [10] introduced a combination of LDA and NLPs for online news media bias. In this paper, the NLPs was used to analyze the adjectives and adverbs because they are usually associated with sentimental bias. Although it does not mention that which NLP methods they were used in this paper, it inspired me to explore this area in the research.

Originally, I had an idea was the combination of text mining methods and topic modeling model. But, in the recently literatures, only the combination of text mining methods and NLP methods is explored. For example, a study was using clustering and classification to fetch the topics. In 2015, Alsmadi and Alhami [11] described algorithms to perform clustering and classification for email contents. Classification based on NGram was the best for large text collection. If the text is Bi-language, it is more effective.

III. METHODS

In the environment of the experiment, Python 3 is used to be the programming language and the Reuters Corpus Volume I (RCV1) [12] contains 50 news collections for testing. In NLP methods, I used count vectorizer and tf-idf vectorizer. In topic modeling model, I chose LDA be the base model.

The experiment tested two combinations of NLP and topic modeling, that are count vectorizer and LDA, and tf-idf vectorizer and LDA.

A. *Latent Dirichlet Allocation*

LDA is published by Blei, Ng and Jordan [4], which is a machine learning and topic modeling technique. The basic idea is documents can be represented by a group of topics, and the feature of each topic is the distribution of words. LDA is a bag-of-words model. It assumes that document is combined by many words. It does not care the order of words.

The algorithm of LDA is:

1. Choose document size *N*
2. Choose a group of topics *T* ~ Dirichlet (α)
3. For each of the *N* words w_n in document *d*:

   (a) Choose a topic t_w ~ Multinomial (*T*)

   (b) Choose a word w_n from P (w_n | z_n, β)

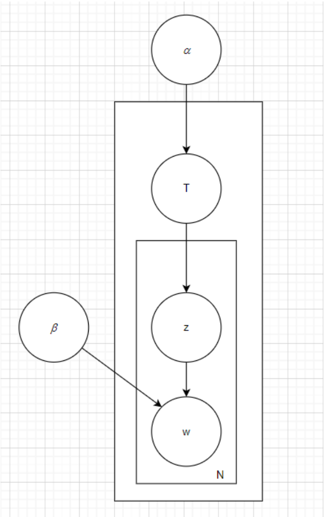


Fig. 1. Structure of LDA

B. *Count Vectorizer*

Count Vectorizer [13] aka one-hot encoding is one of the methods of word embeddings. The concept of count vectorizing is creating a vector that has the same number of dimensions as the unique words of corpus. Each unique word has a specific dimension, only that dimension is 1, other dimensions are 0. The limitation of count vectorizer is the number of vectors is huge and the distribution of vectors are sparse.

C. *Term Frequency Inverse Document Frequency (TF-IDF) Vectorizer*

TF-IDF Vectorizer [13] has the same structure of count vectorizer, but different counting method. TF-IDF vectors counts the term frequency multiplied by the inverse document frequency rather than the number of features. It considers overall documents of weight of words.

The formula of TF-IDF is:

$$\text{TF-IDF} = TF(t, d) * IDF(t) \qquad (1)$$

In here, *t* means word/term and *d* means document. TF is the term frequency which is the number of times of a word appears in a document. IDF is inverse document frequency:

$$log\ (1+n\ /\ 1+df(d,\ t))\ +1 \qquad (2)$$

In the formula of IDF, *n* means the number of documents, and *df(d, t)* means the document frequency of the word/term *t*.

D. Algorithm process of each mashup model

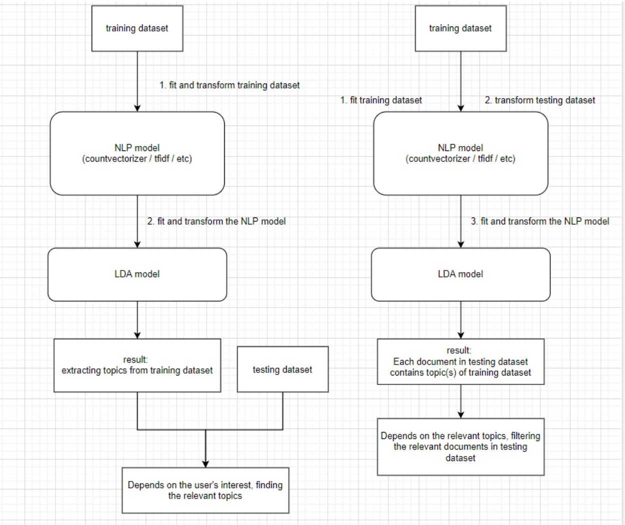

training dataset
1. fit and transform training dataset
NLP model (countvectorizer / tfidf / etc)
2. fit and transform the NLP model
LDA model
result: extracting topics from training dataset
testing dataset
Depends on the user's interest, finding the relevant topics
training dataset
1. fit training dataset
2. transform testing dataset
NLP model (countvectorizer / tfidf / etc)
3. fit and transform the NLP model
LDA model
result: Each document in testing dataset contains topic(s) of training dataset
Depends on the relevant topics, filtering the relevant documents in testing dataset

Fig. 2. Flow chart of algorithm process

In Fig. 2, training dataset and testing dataset represents only one user, each NLP model only executes one NLP method. Training dataset means the user's reading history. Testing dataset means another news collection which is not read by the user. I created two models by using the same NLP method every time, one is only for the training data, another one is for the training data and testing data. The different between these two models is the first one is using to extract the user's interests from the training data and find the relevant topics of the interests, another one is using to label the topics fetched from the training data to the testing data and filtering the relevant news in the testing dataset. The first model is a preparation of the information filtering system, the second model is the execution.

The process of algorithm below:

- Firstly, fetching the training data from the RCV1 dataset and using it to be the parameter of the NLP model 1 for fitting and transforming.
- Secondly, creating a LDA model 1 by fitting and transforming the NLP model 1.
- Thirdly, the LDA model 1 outputs the extracting topics from training dataset and depending on the user's reading history to find the relevant topics.
- Fourthly, building the NLP model 2 by fitting the training data and transforming the testing data.
- Fifthly, building the LDA model 2 by fitting and transforming the NLP model 2. Therefore, the documents of testing data are labelled by the topics extracting from training data.
- Finally, according to the relevant topics, the LDA model 2 filters the relevant news in testing data.

IV. RESULTS AND DISCUSSION

A. Evaluation

To evaluate the novel models, I used few different measurements, such as perplexity, log-likelihood, information filtering evaluation [6] and the accuracy of the relevant news. Perplexity is one of the common measurements in machine learning. It is used to measure the surprised value of the model when it is given a new dataset. It is calculated by the normalized log-likelihood of the held-out test data. The standard of a good model is having a low perplexity.

The formula of log-likelihood is:

$$Log\text{-}likelihood = \sum_D log_2\ p(w_d,\theta)\ /\ \phi \qquad (3)$$

The formula of perplexity is:

$$\text{Perplexity(D')} = 2^{\text{-log-likelihood}} \qquad (4)$$

In (3), D means the document, w_d means the unseen data in the hold out set, θ means the learnt model parameters, and the ϕ means the count of tokens.

The information filtering evaluation is used to find the relevancy of a new document depending on the user's interest by calculating a Score(d). The range of Score(d) is 0 to 1. When the Score(d) is larger, the relevancy is higher. After calculating the Score(d), the system uses the score to make a relevant ranking for the output. If the Score(d) is larger than 0.5, the document is relevant or only the top 10 documents are relevant.

The formula of information filtering evaluation is:

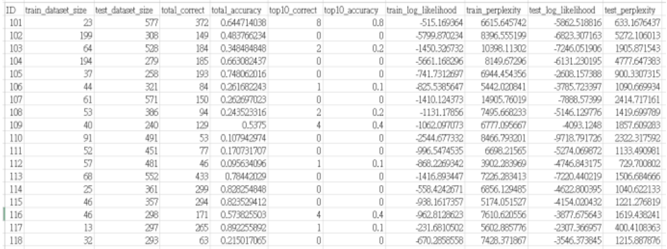
$$\text{Score(d)} = \sum \text{P(w|t)} * \text{P(t)} \qquad (5)$$

In (5), d means document, w means word and t means topic. P(w|t) means the probability of the word w of topic t in document d. For example, a document d is extracted 5 topics and each topic has 3 words. In topic t_1, the probability of word w_2 in d is 0.05. In topic t_2, the probability of word w_4 is 0.12 and the probability of word w_5 is 0.23. Then, P(w|t) is 0.05, 0.12 and 0.23. P(t) is the probability of topic t in document d. For example, according to the user's interest topics, t_2 is the relevant topic but t_1 is not. Assuming $P(t_2)$ is 0.3, the score(d) is (0.12+0.23) * 0.3 = 0.105.

The last measurement is the accuracy of the relevant news. It is used to compare the output of the novel model and the relevant news in testing data.

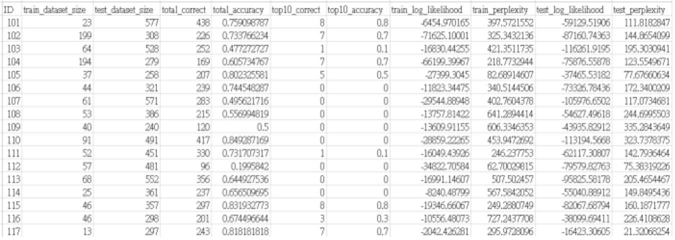
B. Results

I evaluated each LDA model 2 of the novel models. I generated a excel file which contains the ID of user, the size of training dataset, the size of testing dataset, the accuracy of relevant news, the accuracy of the information filtering evaluation ranking, the log-likelihood score and perplexity.



ID	train_dataset_size	test_dataset_size	total_correct	total_accuracy	top10_correct	top10_accuracy	train_log_likelihood	train_perplexity	test_log_likelihood	test_perplexity
101	23	577	372	0.644714038	8	0.8	-515.169364	6615.645742	-5862.518816	633.1676437
102	199	308	149	0.483766234	0	0	-5799.870234	8396.555199	-6823.307163	5272.106013
103	64	528	184	0.348484848	2	0.2	-1450.326732	10398.11302	-7246.051906	1905.871543
104	194	279	185	0.663082437	0	0	-5661.168296	8149.67296	-6131.230195	4777.647383
105	37	258	193	0.748062016	0	0	-741.7312697	6944.454356	-2608.157388	900.3307315
106	44	321	84	0.261682243	1	0.1	-825.5385647	5442.020841	-3785.723397	1090.669934
107	61	571	150	0.262697023	0	0	-1410.124373	14905.76019	-7888.57399	2414.717161
108	53	386	94	0.243523316	2	0.2	-1131.17856	7495.668233	-5146.129776	1419.699789
109	40	240	129	0.5375	4	0.4	-1062.097073	6777.095667	-4093.1248	1857.609283
110	91	491	53	0.107942974	0	0	-2544.677332	8466.793201	-9718.791726	2322.317592
111	52	451	77	0.170731707	0	0	-996.5474535	6698.21565	-5274.069872	1133.490981
112	57	481	46	0.095634096	1	0.1	-868.2269342	3902.283969	-4746.843175	729.700802
113	68	552	433	0.78442029	0	0	-1416.893447	7226.283413	-7220.440219	1506.684666
114	25	361	299	0.828254848	0	0	-558.4242671	6856.129485	-4622.800395	1040.622133
115	46	357	294	0.823529412	0	0	-938.1617357	5174.051527	-4154.020432	1221.276819
116	46	298	171	0.573825503	4	0.4	-962.8128623	7610.620556	-3877.675643	1619.438241
117	13	297	265	0.892255892	1	0.1	-231.6810502	5602.885776	-2307.366957	400.4108363
118	32	293	63	0.215017065	0	0	-670.2858558	7428.371867	-3546.373845	1215.887876

Fig. 3. Results of TF-IDF Vectorizer cross LDA model



ID	train_dataset_size	test_dataset_size	total_correct	total_accuracy	top10_correct	top10_accuracy	train_log_likelihood	train_perplexity	test_log_likelihood	test_perplexity
101	23	577	438	0.759098787	8	0.8	-6454.970165	397.5721552	-59129.51906	111.8182847
102	199	308	226	0.733766234	7	0.7	-71625.10001	325.3432136	-87160.74363	144.8654099
103	64	528	252	0.477272727	1	0.1	-16830.44255	421.3511735	-116261.9195	195.3030941
104	194	279	169	0.605734767	7	0.7	-66199.39967	218.7732944	-75876.55878	123.5549671
105	37	258	207	0.802325581	5	0.5	-27399.3045	82.68914607	-37465.53182	77.67660634
106	44	321	239	0.744548287	0	0	-11823.34475	340.5144506	-73326.78436	172.3400209
107	61	571	283	0.495621716	0	0	-29544.88948	402.7604378	-105976.6502	117.0734681
108	53	386	215	0.556994819	0	0	-13757.81422	641.2894414	-54627.49618	244.6995503
109	40	240	120	0.5	0	0	-13609.91155	606.3346353	-43935.82912	335.2843649
110	91	491	417	0.849287169	0	0	-28859.22265	453.9472692	-113194.5668	323.7378375
111	52	451	330	0.731707317	1	0.1	-16049.43926	246.237753	-62117.30807	142.7936464
112	57	481	96	0.1995842	0	0	-34822.70584	62.70029815	-79579.82763	75.38319226
113	68	552	356	0.644927536	0	0	-16991.14607	507.502457	-95825.58178	205.4654467
114	25	361	237	0.656509695	0	0	-8240.48799	567.5842052	-55040.88912	149.8495436
115	46	357	297	0.831932773	8	0.8	-19346.66067	249.2880749	-82067.68794	160.1871777
116	46	298	201	0.674496644	3	0.3	-10556.48073	727.2437708	-38099.69411	226.4108628
117	13	297	243	0.818181818	7	0.7	-2042.426281	295.9728096	-16423.30605	21.32068254

Fig. 4. Results of Count Vectorizer cross LDA model

According to the results of TF-IDF Vectorizer cross LDA model (tf-idf LDA) and Count Vectorizer cross LDA model (count LDA), the perplexity of tf-idf LDA is 1596.939 and count LDA is 181.6312. The accuracy of relevant news of tf-idf LDA is 51.2911% and count LDA is 67.4783%. The accuracy of information filtering evaluation ranking (top 10) of tf-idf LDA is 15.2% and count LDA is 18.4%. Therefore, count LDA is more suitable than tf-idf LDA to filter the relevant news.

C. Discussion

According to the result of the comparison of tf-idf LDA and count LDA, I understand TF-IDF is not a good NLP method in long textual contents. To improve these models, I think the number of topics is an important variable in NLP methods and LDA. I think different size of dataset should have a different number of topics. It is because in the results, I discover when the size of training dataset and testing dataset has a big range, the accuracy decreases and perplexity increases. Therefore, it means the constant number of components can not satisfy all datasets.

Honestly, I do not satisfy result of this research. Since the time is not enough, I cannot explore more combinations to enhance the information filtering system. For example, LSA is a good choice to be the base model, vect2word is a good choice in NLP methods. Also, the coherence value is an important measurement, however, the sklearn library does not support this function.

V. CONCLUSION

In conclusion, this paper proposes two mashup models, which are tf-idf LDA and count LDA to extract the interest from the user's reading history and filter the relevant news for the user. According to the perplexity and accuracy of each model, I discover count LDA is better than tf-idf LDA for filtering the news collection.

In the future, I will explore different combinations of model and the equation of the best number of topics. Also, the novel model can apply in the recommendation system of newspaper website or library.

REFERENCES

[1] Y. Gao, Y. Xu, & Y. Li, “Pattern-based Topics for Document Modelling in Information Filtering,” IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 6, pp. 1629-1642, 2015. Available: https://doi.org/10.1109/TKDE.2014.2384497

[2] Hanani, U., Shapira, B., & Shoval, P. “Information Filtering: Overview of Issues, Research and Systems,” User Modeling and User-Adapted Interaction, vol. 11, no. 3, pp. 203-259, 2001. Available from: https://doi.org/10.1023/A:1011196000674.

[3] Rania Albalawi, Tet Hin Yeap, & Morad Benyoucef, “Using Topic Modeling Methods for Short-Text Data: A Comparative Analysis,” Frontiers in Artifical Intelligence, vol. 3, 2020. Available: https://doi.org/10.3389/frai.2020.00042

[4] D. M. Blei, A. Y. Ng, & M. I. Jordan, “Latent Dirichlet Allocation,” Journal of Machine Learning Research, vol. 3, pp. 993-1022, May 2003.

[5] D. M. Blei, “Probabilistics topic models,” Commun. ACM, vol. 55, no. 4, pp. 77, April 2012.

[6] D. Kapugama Geeganage, Y. Xu, & Y. Li, “Semantic-based topic representation using frequent semantic patterns,” Knowledge-Based Systems, vol. 216, pp. 106808, 2021. Available: https://doi.org/10.1016/j.knosys.2021.106808

[7] Y.-K. Tang, X.-L. Mao, H. Huang, X. Shi, G. Wen, “Conceptualization topic modeling,” Multimedia Tools Appl., vol. 77, no. 3, pp. 3455-3471, 2017. Available: http://dx.doi.org/10.1007/s11042-017-5145-4

[8] S. Robertson, H. Zaragoza, & M. Taylor, “Simple BM25 extension to multiple weighted fields,” Proc. 13th ACM Int. Conf. Inform. Know. Manag., pp. 42-49, 2004.

[9] Hajjem Malek, & Chiraz Latiri, “Combining IR and LDA Topic Modeling for Filtering Microblogs,” Procedia Computer Science, vol. 112, pp. 761-770, September 2017. Available: https://doi.org/10.1016/j.procs.2017.08.166

[10] Doumit, S., & A. Minai, “Semantic Knowledge Inference from Online News Media Using an LDA-NLP Approach,” The 2011 International Joint Conference on Neutral Networks, pp. 3068-71, 2011. Available: https://doi.org/10.1109/IJCNN.2011.6033626

[11] Alsmadi, Izzat, & Ikdam Alhami, “Clustering and Classification of Email Contents,” Journal of King Saud University. Computer and Information Sciences, vol. 27, no. 1, pp. 46-57, 2015. Available: https://doi.org/10.1016/j.jksuci.2014.03.014

[12] D. D. Lewis, Y. Yang, T. G. Rose, F. Li, “RCV1: A new benchmark collection for text categorization research,” J. Mach. Learn, Res. 5, pp. 361-397, 2004. Available: https://dl.acm.org/doi/10.5555/1005332.1005345

[13] Garreta, Raul., & Guillermo, Moncecchi, “Learning Scrikit-Learn: Machine Learning in Python,” 1st edition, Packt Publishing, 2013.