Inter-topical Keyword Recommendation based on Movie Dialogue Corpus

Zhuoran Li zhl274@ucsd.edu
Yiyi Liang y6liang@ucsd.edu
Xuanyu Chen xuc006@ucsd.edu
Yuxiang Zhou yuz009@ucsd.edu

Abstract

The increase demand of communication efficiency in digital-era encourages data scientists to collect data and develop a system which facilitates the conversation intelligently and emotionally. To achieve such goal, a inter-topical keywords recommendation system is a appropriate solution, which extracts keywords within selected topics, as well as recommends shortest sequential path for leading initial topic to target topic. This paper presents a inter-topical keyword recommendation system to present connection between to input topics. We compared the topic extraction results with different topic models such as LDA, TF-IDF, WordNet, and Word2Vec. The paths related to keywords is calculated using BFS algorithm. We found Word2Vec provides the best results. Our system is able to provide sequential inter-topical keywords for dialogues in business and technology sectors.

1 INTRODUCTION

INTER-TOPICAL keyword recommendation system refers to the process of automatic extraction of essential and topical lexis that relate to and connect topics (e.g., politics, sports, and movie) to automatically bridge between a starting topic and an ending topic. Automatic keyword extraction through topic modeling and graph search can be applied in business (e.g., automated customer service in e-commerce) and technology (e.g., human-machine interaction).

Although human-machine conversation has received much attention across academia and industry in recent years, current dialogue system is still in its infantry. Such system generally interacts with human passively and utters out their responses as reaction to human's input rather than their own initiatives. Automated customer service in the Alibaba e-commerce system, for instance, requires customers to self-select questions from a catalog of possible interests that might or might not cater to customer needs. Such application involves basic information retrieval models, which take human's input as potential topic to retrieve topic-related information and thus leads to limited and inflexible applications. Some other systems provide customized responses through topic extraction and text generation models (e.g., N-gram).

Such passiveness and inflexibility in text generated using the existing dialogue system hamper the development of human-machine interaction, resulting in only weak AI that follows human's instructions. Ideally, however, an automatic conversational AI knows how to proactively lead the conversation by generating utterances that keep the conversation cohesive and informative and link between beginning and ending topics. Although this ideal prospect of developing such a powerful AI seems ambitious in the near future, the first step is practically attainable. The present study proposes an inter-topical keyword recommendation model that identifies keywords related to bridging topics between the input topics and facilitates automatic inter-topical conversation generation.

The present study employs four models/techniques for topic extraction: **Latent Dirichlet Allocation** (**LDA**), **Term Frequency–Inverse Document Frequency** (**TF-IDF**), **WordNet**, and **Word2Vec**. Previous works related to the four models are studied and referenced for the present study and will be introduced in the **Chapter 2**.

In this paper, we propose a topic-keyword extraction method to find the connection of two topics. In this method, a collection of keywords is extracted to represent the shortest sequential path between two topics. The path between topics provides a idea to lead to topic B from topic A. We analyzed the corpus from Cornell Movie-Dialogs dataset, and built up a word net. Based on the graph, we applied Breadth-first search to calculate the number of nodes. We connected two topic by recommending the path which contains lowest number of nodes.

The present study consists of eight chapters:

1. **Introduction** gives an overview of the background of topic modeling and text generation as well as the main components of the present study.

2. **Related Work** reviews some previous works that use the topic models for topic recommendation and text generation.

3. **Datasets** specifies the source for the data used in the present study.

4. **Prelimiaries** introduces basic algorithms and concepts that the following study uses.

5. **Methodology** explains data pre-processsing and four different approaches to extract topics and keywords from corpus and find the shortest path between input topics.

6. **Experimental Results** exhibits keywords extracted from topic combinations with examples.

7. **Conclusion** summarizes the experimental findings.

8. **Discussion** discusses the present study's strengths, weaknesses, potential improvements.

2 RELATED WORK

Prior to the present study, mainstream topic models, including TF-IDF, LDA, WordNet, and Word2Vec have been widely adopted by academia and industry for topic extraction and recommendation system, especially in social media-based corpus analysis. The present chapter reviews methodologies adopted by previous studies and projects that lend insights for the present study.

2.1 TF-IDF

The most similar recommendation system to the topic recommendation is tag recommendation. Vairavasundaram proposed a data-mining based tag recommendation system [16]. Generated from user content, the tags help annotate the related content and recommend to other users.

The use of Natural Language Processing (NLP) techniques in recommendation system can be manifold, but TF-IDF is definitely among the most effective and easy-to-implement methods for topic extraction in social media-based corpora. Tajbakhsh introduced Twitter hashtag recommendation system based on TF-IDF in 2016[14]. The recommendation system is adapted to Twitter posts, which contain shorter massages than regular blogs. Cosine similarity is used to determined to similarity of the words.

2.2 LDA

Blei et al. first introduced Latent Dirichlet Allocation (LDA) model in 2003[2]. LDA has been widely used in topic modeling in various domains. Godin et al. discovered hidden topics and developed a hashtag recommendation system with LDA using Gibbs sampling[7]. She and Chen built a TOHOMA system, a supervised model based on LDA for recommending hashtags in Twitter[12]. The supervised model can infer the hashtags based on the relationships among them. Apaza et al. proposed a online course recommendation system for online education industry[1]. Based on topic as well as grading information, the content-based recommendation system provided relevant courses to students.

LDA is applied to other related functional area as well. Liu et al. developed system to detect spammers on social networks [9]. The system captures the spammers using the information of topic distribution patterns. A topic model combining topic recommendation as well as domain-specific knowledge is used in political science field[8]. Analyzing the UK House of Commons speeches, they developed a semi-automatic method to transfer the topic labels.

2.3 WordNet

Wordnet is also applied to the recommendation system as well [13]. Choi et al. provided a movie genre recommendation system using synonyms from WordNet [4]. Genre correlations are drawn using the counting method. Capelle et al. applied the Wordnet to the news recommendation system[3]. In the system, WordNet synonym sets are applied to calculate the similarity of the news, with dataset from Bing Search Engine.

Although topic models are applied to various domain, few articles are related to recommend two topics with sequential path as well as the linking keywords.

2.4 Word2Vec

Word2Vec is a family of model architectures and optimizations that can be used to learn word embedding from large corpus datasets. It is firstly developed by Mikolov et al.[10]. Embedding technique helps project words(or items) to a k-dimension space.

The most frequent Word2Vec algorithms are skip-gram and CBOW models, using either hierarchical Softmax or Negative sampling to produce word embedding with vector space and transfer corpus to vectors for machine learning and neural network.

Once trained, Word2Vec represents each distinct word with a particular vector. The vectors are chosen carefully based on simple metrics like cosine similarity which indicates semantic similarities between the words represented by corresponding vectors.

3 DATASETS

The present study uses the Cornell Movie-Dialogs Corpus dataset to test and validate the effectiveness of model features. This dataset was created from the IMDB database by Danescu [5]. The dataset contains corpus of raw movie scripts in fictional conversations as well as related metadata. The corpus includes 220,579 conversational exchanges between 10,292 pairs of characters in 617 movies. The total utterances are 304,713. A collection of metadata including genres, release year, IMDB rating, number of IMDB votes, character gender, and position on movie credits are in the dataset as well.

4 PRELIMINARIES

The present study proposes several keyword extraction solutions based on the concepts of **Cosine Similarity**, **Term Frequency-Inverse Document Frequency** (**TF-IDF**) and **Latent Dirichlet Allocation** (**LDA**). The present chapter introduces these concepts and methods as preliminaries for the present study.

4.1 Cosine Similarity

Cosine Similarity is a commonly-used measure of similarity between two vectors by the cosine of the angle between them. Given vectors $A, B \in \mathbb{R}^d$, the cosine similarity between A and B is defined using a dot product and norms as:

$$\begin{aligned} CosSim(A,B) &= \frac{A \cdot B}{\|A\|\|B\|} \\ &= \frac{\sum_{i=1}^{d} A_i \times B_i}{\sqrt{\sum_{i=1}^{d} A_i{}^2} \times \sqrt{\sum_{i=1}^{d} B_i{}^2}} \end{aligned}$$

The definition of Cosine Similarity shows that its ranges within $[0, 1]$, where its value indicates the level of similarity between the vectors.

4.2 Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a widely accepted and adopted method for word embedding and is extensively used in the information retrieval area. The term of TF-IDF consists of two parts, Term Frequency and Inverse Document Frequency. It therefore measures both how frequent a term appears in a single document and how frequent it appears across multiple documents. Given a term t, a document d, and a corpus consisting of multiple documents D, the raw form of TF, IDF, and TF-IDF are defined as:

$$TF(t,d) = f(t,d)$$

where $f(t,d)$ is the raw frequency of t in d

$$IDF(t,d,D) = \frac{|D|}{|d \in D; t \in d|}$$

$$TFIDF(t,d,D) = TF(t,d) \times IDF(t,d,D)$$

to which log-transformation and smoothing can be applied in practice. Then, transform d into a TFIDF-weighted vector given a corpus D and a vocabulary $T = \{t_1, t_2, \dots, t_n\}$ as:

$$\begin{aligned} &TFIDF(d,D) = \\ &\langle TFIDF(t_1,d,D), \dots, TFIDF(t_n,d,D)\rangle \end{aligned}$$

Hence, the cosine similarity between two documents d_1 and d_2 is calculated as:

$$\begin{aligned} &DocCosSim(d_1,d_2,D) = \\ &CosSim(TFIDF(d_1,D), TFIDF(d_2,D)) \end{aligned}$$

Now that the cosine similarity between two documents is defined, TF-IDF can be used for topic and keyword extraction.

4.3 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is a generative model that allows a document to be represented by a mixture of topics[15]. The idea behind LDA is that the author of a document has a set of topics in mind when writing the document, and thus a topic can be represented by a collection of terms (e.g., keywords) with relavant semantic meaning. The LDA model is formally described as follows:

$$P(t_i|d) = \sum_{j=1}^{|Z|} P(t_i|z_i = j) \cdot P(z_i = j|d)$$

where:

- $P(t_i|d)$ denotes the probability of term t_i being in document d
- z_i is a latent (potential) topic
- $|Z|$ is the pre-defined total number of topics
- $P(t_i|z_i = j)$ denotes the probability of t_i being in topic j
- $P(z_i = j|d)$ denotes the probability of picking a term from topic j in document d.

Similarly, LDA can be used to represent a document by a vector of probabilities of the document belonging to a topic, using the *inference* technique:

$$Infer(d, Z) = \langle z_1, z_2, \ldots, z_T\rangle; T = |Z|$$

Again, use cosine similarity on inferences to measure similarity between two documents d_1 and d_2 as follows:

$$InfCosSim(d_1, d_2, Z) = CosSim(Infer(d_1, Z), Infer(d_2, Z))$$

4.4 Breadth-First Search Algorithm (BFS)

Moore first invented Breadth-First Search algorithm in 1959[11]. BFS algorithm starts from a vertex as a search key, and discovers all the neighbour nodes prior to depth nodes to find the target node. A sample graph representing the nodes is shown in Figure 1[6].

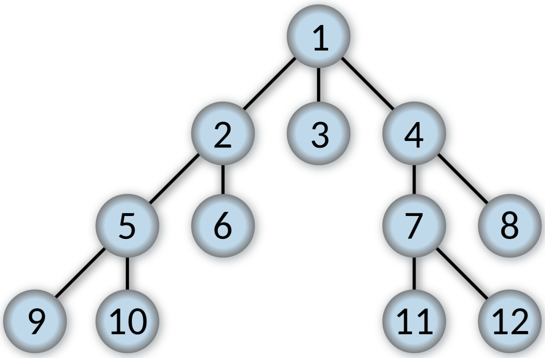

Figure 1: Breadth-First Search Tree

5 Methodology

5.1 Data Processing

Data processing is mainly used in two parts: pre-processing the (raw) movie lines and process the similar words got from Google pre-trained Word2Vec model(word2vec-google-news-300). Here are the steps of data **pre-processing**:

1. **Remove punctuation**: Using package **re** is a good way to remove punctuation and other special characters
2. **Tokenization, Lemmatization or Stemming**: if using PorterStemmer() function from **nltk** package, the result of stemming word is always hard to understand, e.g., the word ”something” will become ”somth” after stemming. On the contrary, ”en_core_web_sm” from **Spacy** is a good way to get words' lemma.
3. **Extract nouns**: two databases from **nltk** package are used in this part which are **WordNet** and **averaged_perceptron_tagger**. As for the tokens not in **WordNet** or not targeted as NN by **averaged_perceptron_tagger**, they will not be selected. In the end we choose to use **WordNet** to filter out nouns.

As for every unique noun, there are 10 similar words got from Google pre-trained Word2Vec model(word2vec-google-news-300). Here are the steps of **processing both of original words and their corresponding similar words**:

1. **Transforming the original words to their corresponding lemma form**
2. **Removing repeated words**: As for the 10 similar words, removing the ones which have

repeated with other similar words or that are repeated with the original word.

5.2 TF-IDF model

A word pair with the highest two TF-IDF scores is extracted from each line as the topics from that line. We label these two words as related topics and mark a edge between them. Two different TF-IDF models are used in topic extraction. The first model extract topics for each line with only considering the current movie data. The second model extract topics for each line based on the entire data-set from over 600 movies.

5.3 Latent Dirichlet Allocation

gensim.models.ldamodel.LdaModel class from **Gensim** is a great way to implement LDA model. This module allows both LDA model estimation from a training corpus and inference of topic distribution on new, unseen documents. The model can also be updated with new documents for online training.

Here are some important parameters of LdaModel class in practice: the input **corpus** parameter of LdaModel() class is an nested list in which every sublist contains the tokens processed from each movie line; As for input parameter **id2word**, it is a dictionary mapping from word IDs to words which is used to determine the vocabulary size, as well as for debugging and topic printing; **num_topics = N** represents the number of topics we choose to extract from corpus.

When the LDA model is built and trained on the movie liens corpus, it offers a probabilistic distribution of N topics which contains 20 tokens for every topic. For example, one of the topics is: [(20, '0.697*"think" + 0.126*"night" + 0.121*"ask" + 0.032*"party" + 0.016*"date" + 0.006*"saturday" + 0.000*"thin" + 0.000*"port" + 0.000*"filthy" + 0.000*"prime" + 0.000*"minister" + 0.000*"berlin" + 0.000*"willie" + 0.000*"india" + 0.000*"disease" + 0.000*"soil" + 0.000*"tick" + 0.000*"curtain" + 0.000*"shake" + 0.000*"carbon"')]. It means that topic #20 is composed of "think", "night", "ask", etc. and the number before every token represents the probability that this specific token belongs to this topic.

In experiment, several numbers of topics are tried. Firstly **num_topics** is set to be 500 but the outcome is bad since more than four hundred topics are identical. It maybe due to the lack of corpus. When **num_topics** is set to be 100 then the outcome is appropriate. Therefore, we finally choose 100 as the number of topcis

5.4 Word2Vec

We use the unique nouns from the movie data-set as the seed topics and get the top 10 similar words for each of them with the pre-trained word2vec-google-news-300 model. We define each noun and their top 10 similar words as related topics

5.5 Edge Detection and Graph Building

We have different edge building algorithms:
TF-IDF: For each movie line, we extract the top two topics and mark them as related topic pair.
LDA: We extract the top two topics from each line and find the top two words that represent each topic. We mark all combinations between these 4 topics as related topics pairs.
Word2Vec: We define each unique noun and it's top 10 similar words as all related. So each pair combination of these 11 words are marked as topic pair.
For each topic pair, we make an non-direction edge between them and build a graph with all edges.

5.6 BFS

BFS is implemented with queue to traverse the given graph. If the shortest path between the beginning topic and the ending topic is found, the BFS will output the shortest path as the best way to connect the given topic pairs, if it exist.

6 Experimental Results

We extract the following move lines:

Movie Line 1: 'Thank God! If I had to hear one more story about your coiffure...'

Movie Line 2: "Who knows? All I've ever heard her say is that she'd dip before dating a guy that smokes."

Movie Line 3: "Unsolved mystery. She used to be really popular when she started high school, then it was just like she got sick of it or something."

Algorithm	Output
TF-IDF-1	story (1.95), god (1.64)
TF-IDF-2	story (1.63), god (1.63)
LDA	hear, word, start, god

Table 1: Topic keywords and for movie line 1

Table 1, table 2 and table 3 show the extracted topics from different algorithm. TF-IDF-1 is

Algorithm	Output
TF-IDF-1	dip (1.91), dating (1.71)
TF-IDF-2	dip (1.16), dating (1.02)
LDA	guy, understand, think, night

Table 2: Topic keywords and for movie line 2

Algorithm	Output
TF-IDF-1	mystery (1.66), high (1.24)
TF-IDF-2	sick (1.16), mystery(1.16)
LDA	love, use, something, put

Table 3: Topic keywords and for movie line 3

the TF-IDF algorithm that take the entire 617 movie data and TF-IDF-2 is the TF-IDF algorithm that consider each movie individually. Although the topics that these two algorithm extracted as highly similar, topics from TF-IDF-1 have more weight leaning to that movie line. Comparing topics from TF-IDF and LDA, TF-IDF can capture the core topics in the movie lines better.

Here are the results of shortest paths got from different models:

TF-IDF-2:

Topic Keywords	**Shortest path**
flower, box	flower → bed → box
bread, class	bread → run → class
class, doctor	class → today → doctor

Table 4: Topic keywords and shortest path based on TF-IDF

The TF-IDF-2 is selected to represent the TF-IDF track. The searching time for BFS is generally less than 10 seconds and the recommended shortest path is usually very short. From the examples in Table 5, we can see some intuitive connections in the path. For (bread, class) pair, the path can be understood as ”eating bread” while ”running” because ”the class is starting soon”. However, this indeed need extra human imagination to make up the story.

LDA:

As for every movie line, LDA model can give a probabilistic distribution of 100 topics in which every topic is composed of 20 words. During the experiment, every movie line is summarized by four words which are the 2 most likely words in each top 2 topics in order to build a graph. But just a few start/end words can be chosen from this graph and the path between starting word and end word will be extremely short.

Topic Keywords	**Shortest path**
lunch, country	lunch → country
look, guy	look → minister → guy
play, game	play → game

Table 5: Topic keywords and shortest path based on LDA

Word2Vec:

Topic Keywords	**Shortest path**
bread, car	bread → flour → wheat → wheat_crop → grower → tractor → car
math, marriage	math → mastery → immortality → nirvana → marital_bliss → marriage
dog, homework	dog → feline → kitty → piggy_bank → checkbook → payroll → bookwork → homework

Table 6: Topic keywords and shortest path based on Word2Vec

The Word2Vec model preforms significantly better than LDA and TF-IDF. The intuitive connection between each path is clear. Most of the pairs can result a path in seconds while occasionally the BSF takes more than 3 minutes to run. However, sometimes the model overthink the connection path between the beginning words and ending word and might miss some common phrase. With the (dog, homework) example, the model makes a path from dog to kitty, to checkbook, to payroll, bookwork, and then homework. With some intuitive, a human might make a simpler path as ”dog”, ”eat”, ”homework” from jokes.

Algorithm	Nodes	Edges
LDA	200	4209
TF-IDF-2	17815	116499
Word2Vec	351359	406886

Table 7: number of nodes and edges

The LDA and TF-IDF-2 are giving short paths since the node-edges ratio is low. The ratio for Word2Vec is more reasonable.

7 Conclusion

This paper presents a system for inter-topical keyword to create a collection of sequential keywords. Using the LDA, TF-IDF, WordNet, and BFS algorithms, we extracts the keywords of the movie dialogues dataset and create a network with nodes. By comparing the results, the best model is the graph built from Word2Vec. For most word pairs, it successfully maps paths from beginning words to ending words with a reasonable run-time.

8 Discussion

8.1 Limitation

The topic-keyword extraction model generates a collection of keywords with sequential manner. There are some limitations in this model. For example, the word-net is created by Breadth-first search. The words of some word-nets are too close to provide effective sequential paths. Although a path with few nodes represents simple sequence to change the topic, the relationship between topics may not detected comprehensively. Therefore, further research for advanced algorithms is necessary.

The dataset used in the study contains only movie dialogues. Those corpus are limited by the movie characteristic and genre. Limited corpus affects the application of the results as well as the capability. Moreover, the dataset was collected in 2011, which does not contain the latest corpus with their novel connections. Therefore, A dataset with larger scope and latest recency is needed for further research.

The evaluation of the model results are based on individuals. Manual labeling could be achieved by domain experts, or even by the automatic labeling evaluation system with advanced algorithms.

One improvement could be achieved is the processing time. If the path of two words is simple and straightforward within in net, the process can be done in 10 to 20 seconds. However, if the path between two words are long and complicated, the processing time with be too large and sometimes cannot achieve the results. The algorithm could be improved by setting maximum running time or the early stopping rules. Also, we can substitute the topic words with their synonyms to compare the synonyms and recommend the pairs with lowest processing time.

After LDA model is trained on the movie conversation corpus, every movie line has its corresponding probabilistic distribution of 100 topics in which every topic is composed of 20 words, but it is hard to summarize every movie line by k words due the limitation of using graph.

8.2 Future work

In this topic-keyword extraction model, Breadth-first search (BFS) net is used to determine paths within the word-nets. The net could be improved by adding weights to each path. A further method called Dijkstra can achieve the weight-based path evaluations. Another improvement could be using bi-direction algorithm which allows topic keyword recommendation can be searched in both directions.

We compare TF-IDF, LDA, WordNet, and Word2Vec in the analysis. There are various models based on the previous models can be applied to the system as well. Those topics can better process the corpus and overcome the disadvantages of current models.

The dataset in this paper, which contains movie dialogues corpus, can be substitute to datasets with larger scope and volume.

The feature for the topic connection can be further developed. For example, collection of words can be transformed to set of sentences with logical or even sentimental consideration. Furthermore, the keywords as well as sentences will updated as the number of previous sentences increase.

References

[1] Rel Guzman Apaza, Elizabeth Vera Cervantes, Laura Cruz Quispe, and José Ochoa Luna. Online courses recommendation based on lda. In *SIMBig*, pages 42–48. Citeseer, 2014.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[3] Michel Capelle, Frederik Hogenboom, Alexander Hogenboom, and Flavius Frasincar. Semantic news recommendation using wordnet and bing similarities. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 296–302, 2013.

[4] Sang-Min Choi, Da-Jung Cho, Yo-Sub Han, Ka Lok Man, and Yan Sun. Recommender systems using category correlations based on wordnet similarity. In *2015 International Conference on Platform Technology and Service*, pages 5–6. IEEE, 2015.

[5] Cristian Danescu-Niculescu-Mizil and Lillian Lee. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*, 2011.

[6] Alexander Drichel. *Breadth-first-tree*. 2008.

[7] Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 593–596, 2013.

[8] Alexander Herzog, Peter John, and Slava Jankin Mikhaylov. Transfer topic labeling with domain-specific knowledge base: An analysis of uk house of commons speeches 1935-2014. *arXiv preprint arXiv:1806.00793*, 2018.

[9] Linqing Liu, Yao Lu, Ye Luo, Renxian Zhang, Laurent Itti, and Jianwei Lu. Detecting" smart" spammers on social network: A topic model approach. *arXiv preprint arXiv:1604.08504*, 2016.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[11] Edward F Moore. The shortest path through a maze. In *Proc. Int. Symp. Switching Theory, 1959*, pages 285–292, 1959.

[12] Jieying She and Lei Chen. Tomoha: Topic model-based hashtag recommendation on twitter. In *Proceedings of the 23rd international conference on World Wide Web*, pages 371–372, 2014.

[13] V Subramaniyaswamy and S Chenthur Pandian. Effective tag recommendation system based on topic ontology using wikipedia and wordnet. *International journal of intelligent systems*, 27(12):1034–1048, 2012.

[14] Mir Saman Tajbakhsh and Jamshid Bagherzadeh. Microblogging hash tag recommendation system based on semantic tf-idf: Twitter use case. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pages 252–257. IEEE, 2016.

[15] Suppawong Tuarob, Line C Pouchard, and C Lee Giles. Automatic tag recommendation for metadata annotation using probabilistic topic modeling. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, pages 239–248, 2013.

[16] Subramaniyaswamy Vairavasundaram, Vijayakumar Varadharajan, Indragandhi Vairavasundaram, and Logesh Ravi. Data mining-based tag recommendation system: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(3):87–112, 2015.