Pre-processing Movie Synopses for LDA

Abstract—**This paper explores the effect of text preprocessing on movie synopses using LDA to infer genres of movies. Literature mostly treats pre-processing of text as a mandatory step before applying LDA but has not sufficiently explored the effect of pre-processing. Specifically, there isn't detailed research on the consequence of different pre-processing techniques on movie synopses before being trained on LDA. To fill the gap in the field, this paper aims to explore feasible pre-processing methods for movie synopses. This paper concludes that tf-idf is a good option for exploring more granular topics, in other circumstances using simple count vector as weighting technique is sufficient. It also seems that adjusting document frequency yields better topic inference. And finally, adding genre specific common words to stop words list improve topic inference quality.**

Keywords—LDA, tf-idf, pre-processing, topic coherence, stop words, movie synopsis, NLP, n-grams, perplexity

I. Introduction

Pre-processing is considered a necessary preparation on text data in Natural Language Processing (NLP). The purpose is to clean data and select features. Denny and Spirling (2018) observe that pre-processing advice have found their ways into textbooks for NLP and have been followed by scholars who apply similar steps without thorough understanding of their text at hands, resulting in possibly incorrect inclusion of features and wrong emphasis on feature weights. When such features are fed into topic models, the results could only be skewed. A more serious consequence is that latent topics are not picked up by models. In the case of political science or other research area, it might lead to researchers judging that a certain topic is important while in fact it might not be the case.

This is true for movie synopses, which are one of those corpora that are accessible, easy to understand without expert knowledge. There are a lot of attempts on the application of topic modelling on movie synopses or reviews for improvements of movie recommendation systems, thanks to the rising popularity of streaming networks such as Netflix, Hulu, Amazon Prime Video.

To fill the gap in the field, this paper aims to explore feasible pre-processing matric for movie synopses. This paper considers the following pre-processing techniques: (1) stop words removal, (2) punctuation removal, (3) stemming, (4) lower-casing, (5) lemmatising. In addition, I consider the compilation of stop words list, adjusting weights, screening terms using minimum and maximum document frequency, the inclusion and exclusion of bi-gram and tri-gram.

The paper focuses on Latent Dirichlet Allocation (Blei et al., 2003). Since LDA was proposed, it has made exploration of textual data in large size (millions of documents) possible across a variety of subjects such as political science, history and literature (Schofield et al., 2017). However, the effect of text pre-processing on the performance of LDA with movie synopsis as input data is not sufficiently explored. One reason is that research has been focused on training LDA itself by tuning hyperparameters. Another reason is that most paper use structured text data, such as New York Times articles, Reuters newswires, or Wikipedia articles. Such datasets are well written, have clear themes, and grammatically structured, therefore ideal for LDA model with simple pre-processing steps. In addition, research has favoured movie reviews combining ratings to be applied with LDA for classification or recommendation system.

The paper uses the MPST corpus, a corpus of over 14k movie synopsis with 71 tags. The movie synopses were combined and selected from Wikipedia and IMDb. The tags were manually selected by the authors of the paper (Kar, 2018) and had been fine grained to fit the plot of the movies. I chose this dataset because the tags would serve as the "ground-truth" labels to compare with the model outcome (Fig 1 illustrates the data collection process of the corpus).

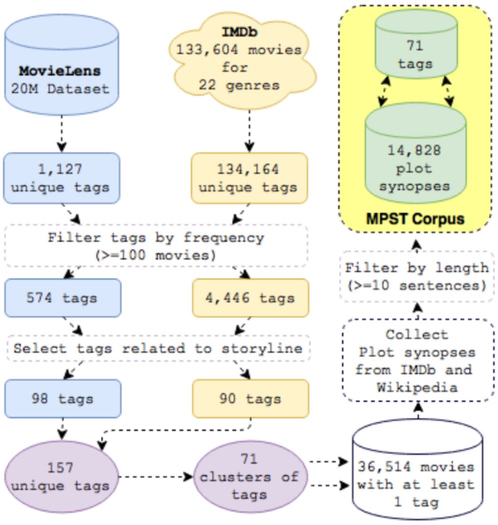


Fig 1: Data collection process of the MPST corpus (Kar et al., 2018)

This paper is structured as follows. Section 2 describes related research and suggestions on pre-processing techniques. Section 3 describes pre-processing techniques, weighting techniques, LDA model, and implementation. Section 4 analyses the results. Conclusion and future work are presented in Section 5.

II. Background Research

Previous papers on LDA pre-processing are not many (Schofield, Magnusson and Mimno, 2017; Schofield and Mimno, 2016; Schofield et al., 2017). Schofield et al (2017) suggests that compiling stop words list is time consuming and has little upside for topic models to learn topics over other terms than stop words. They conclude that instead of removing a list of stop words from the corpus before training, removing very frequent unwanted terms or stop words after training is sufficient.

In another paper dedicated to pre-processing for LDA, Schofield et al (2017) further stated that stemming could be redundant and even has negative impact on the results of models because LDA tends to consider words of the same morphological roots under the same topics.

This paper will test the stop list compilation method as suggested by Schofield et al to see if the above conclusions is valid for the dataset under discussion, and perhaps, as an indication for dataset with similar characteristics: unstructured, without special or strong indicative keywords to infer a prominent theme, has multiple latent topics implied in the plots that could be recognised by a human viewer. The challenge of movie synopses for LDA lies in that a movie could have seemingly contradictory themes, it could be a romantic love story that is also violent such as *Natural Born Killers* and *Bonnie and Clyde*. When reviewing the keywords assigned by the model as associated with a topic, human judgement is required. If a topic is related to words such as 'love' and 'kill' at the same time, should we classify the movie with this dominant topic as romantic or violent? Another characteristic of movie synopses is that there are a lot of verbs. Verbs associate events with characters' actions and form one or several story lines. Synopses use simple and short words. This means that some words are commonly used in all synopses, such as 'find', 'see', 'come', 'go'. As later discovered in the result section, such words predominate almost all topics in the corpus. Therefore, compiling common words list post training the model is required.

Depending on the genres of movies, the language used is different. It is expected that action movie synopses involve a lot of verbs, suspence movie plots will have a mixture of adjetive and verbs, romantic movies might use a lot of adjectives and nouns to indicate the emotional changes of characters. Over 70% of the movies in the corpus selected are assigned as either 'Murder' or/and 'Violence' (a movie has one or more tags, see Fig 2 for the corpus statistics). This indicates that the corpus will have a lot of vebs. As later shown in the model pre-processing step, the number of verbs is so prominent that some could be viewed as stop words.

Total plot synopses	14,828
Total tags	71
Average tags per movie	2.98
Median value of tags per movie	2
STD of tags for a movie	2.60
Lowest number of tags for a movie	1
Highest number of tags for a movie	25
Average sentences per synopsis	43.59
Median value of sentences per synopsis	32
STD of sentences per synopsis	47.5
Highest number of sentences in a synopsis	1,434
Lowest number of sentences in a synopsis	10
Average words per synopsis	986.47
Median value of words per synopsis	728
STD of words per synopsis	966.16
Highest number of words in a synopsis	13,576
Lowest number of words in a synopsis	72

Fig 2: Statistics of MPST corpus (Kar, 2018)

III. Methodology

A. Pre-processing techniques

1) Preparation:

Three techniques will be default implementation before other techniques are applied: lower casing, tokenization, and lemmatization.

2) N-grams:

Though n-gram has been widely included as as a feature of text data for NLP or many language models, the reason of using only unigram (a single word token), or with a mixture of bigram (a sequence of two adjacent tokens) or n-grams (a sequence of n adjacent tokens where n is an integer) in a model needs to be justified as the computation cost is higher while the benefits are unclear. As Denny and Spirling (2018) pointed out, researchers following previous work might not necessarily provide justification for using certain techniques.

First of all, we need to consider if the thought of including n-grams in the model is worthy. LDA is based on the bag-of-words assumptoin, meaning the order of words in a document can be neglected (Blei et al., 2003). Blei et al. stated that the limitation of such assumption is that words that should be generated by the same topic are allocated to various topics. They suggested that LDA could perform well even when the assumption is partially relaxed, e.g. by allowing Markov chain of word sequences. This means that n-grams as features of LDA models in general are a valid consideration.

Next, we look at whether the corpus at hand requires the input of n-gram. For legal documents, using bigrams or trigrams makes sense. For example, cases named after the defendant and the offender for future reference, as in "'Roe V. Wade'" is a meaningful unit worth counted (Denny and Spirling, 2018). For movie synopses, such specialized combination of words would be rare. We generally expect movie synopses to be written in plain language wihout any expert knowledge of what happens in a movie. It is therefore reasonable to assume that n-grams would create noise if added as feature input to the topic model. This seems to be the case when bigram and trigram of the text are experimented. The results of the model showed no obvious indication of the benefit of including n-grams as features for the purpose of this paper.

3) Stop words removal

Stop words are words to be removed in the pre-processing step. There is no clear definition for stop words and not a universal list. The assumption for stop words removal is that some words of very high frequency in a corpus might hinder model performance and thus needs to be removed. Words such as 'the', 'a', 'there', 'one' provide little value to language model. Most libraries have its own standard stop words list. Researchers often download the standard lists and remove them from their corpus, or they compile their own special stop words list.

Since every corpus is unique, this paper sets out to test if the standard stop words removal, that is, importing online existing lists, benefits the inference of topics. Natural Language Toolkit (NLTK) standard stop words list has 179 words. spaCy stop words list has 326 words. After the default pre-processing steps (i.e. lowercasing, tokenizing, lemmatizing), the corpus has 89,314 unique tokens. The number of unique tokens reduces only a few after removing the NLTK stop words, to 89,186 tokens, and after removing the spaCy stop words, to 89,039 tokens. The difference is neglectable. This seems to correspond to the claim by Schofield et al (2017) that stop words removal before training adds little value for model improvement.

Stop words could also be the most common words in a corpus. Following the suggestion by Schofield et al (2017), this paper compiles common words lists post initial training. The common words will be added to the stop words list and removed before the next training.

4) *Document frequency adjustment:*

Low frequency words are noise in a corpus. They increase the dimension of features while adding little value as input. The corpus has many words that has 0 count in most documents and semantically they are not meaningful. Removing low frequency words is therefore a feature selection as well as data cleaning process. The methods to remove low frequency words include setting cut-off for length of words (e.g. removing words whose length is less than 3), cut-off for term frequency (e.g. removaing words that has less than 100 counts in the corpus), and cut-off for document frequency (e.g. removing words that appear in less than 20 documents or in more than 10% of the corpus). As later experiment shows, this technique could be combined with weighting technique to reduce the dimension of features.

B. *Weighting technique: TFIDF*

Term frequence-inverse document frequency, or tf-idf, is a numerical statistic in the field of Informtion Retrival to reflect the importance of a word to a corpus. Traditional document term matrix counts the number of times a term appears in the corpus. The result is that words such as ‘the’ will have far more important weight than words such as ‘psycho’. The tf-idf considers a term that appears in a lot of document less important, or giving less new information, than a term that appears in a few documents (making the document distinct from others). The tf-idf does this by inverting the document frequency when calculationg term weights. Higher document frequency yeilds lower inverse document frequency. This is multipled with tf (term frequency), to give the final weight for a term.

Over the years, the transforming document term matrix using the tf-idf has become one of the most popular techniques in NLP. In the paper that proposed LDA, Blei et al (2003) stated that the assumptions of exchangeability (i.e. word orders and document orders are neglectable) “do not necessarily lead to methods that are restricted to simple frequency counts or linear operations”. This paper will compare the use of simple count and the tf-idf weighting technique for the document term matrix of the corpus.

C. *Latent Dirichlet Allocation*

Latent Dirichlet Allocation is a probabilistic model that could be viewed as distribution over distribution. Consider a collection of documents D. LDA treats D as distributions of documents over topics, and each document as distributions of topics over words. In practice, the number of topics is unknown to the model and will be set manually by human. It greatly influences the inference of topics and words associated with the topics. A collection of documents (the corpus) shares the same group of topics (e.g. K topics). Traditional document mixture models associate each document with a single unknown topic (Mcauliffe and Blei, 2008). Topic models are flexible in that it sees each document as having its unique mixture of topics. In practice in a LDA model, a document will most likely have the same set of topics as another document. The difference between the documents lies in the probability of each topic in a particular document. For example, in LDA, document1 is represented as {topic1, 0.8; topic2, 0.15; topic3, 0.5}; document2 is represented as {topic1, 0.3; topic2, 0.5; topic3, 0.2}.

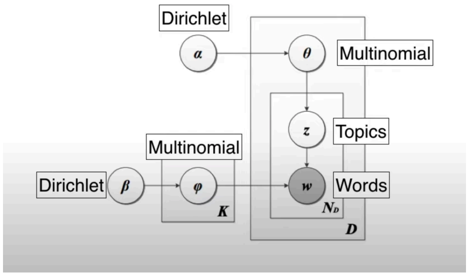


Figure 4: Blueprint of LDA (Serrano, 2020). α and β are Dirichlet distributions. θ and φ are multinomial distributions. z is a topic for a chosen word in a document. w refers to specific words in N, plate D is the length of documents, and plate N is the number of words in the document (Albalawi, Yeap and Benyoucef, 2020).

The only variables that a standard LDA model knows are the words appearing in a document in a collection of documents. The other variables are all inferred, hidden, or latent. Three variables or hyperparameters are important for LDA. One variable is the number of topics. A topic is assigned to a word, making a document a mixture of topics. The other two important inferred variables are α (a Dirichlet distribution, responsible for per document topic distribution) and β (a Dirichlet distribution, responsible for per topic word distribution). Calvo - González, Eizmendi and Reyes (2018) reported that high alpha means an average document is most likely to contain a mixture of most of the topics, low alpha value means a document may contain a few of the topics. High beta means each topic is more likely to contain a mixture of most of the words and a low value of beta means a topic may contain only a few of the words. High alpha makes documents similar to each other; high beta makes topics similar to each other.

D. *Experiment*

The experiment is designed as follows. The two Dirichlet distributions in the LDA model, α and β, are set at fixed values for comparability purpose across different pre-processing techniques. Griffiths and Steyvers (2004) found that $\alpha = 50/t$ (where t is the number of topics of the corpus) and $\beta = 0.01$ work well with different types of text collections.

Different values of α and β are experimented. For this specific corpus, the configuration suggested by Griffiths and Steyvers is not optimal. The resulting topic distribution is indistinct, with average topic distribution over words being 0.05. This paper will use the default symmetric prior (1/number of topics) for the Dirichlet distributions. However, the adjustment of these two hyperparameters help to produce a list of common words that are assigned as keywords associated with the majority of topics over and over again by LDA across different hyperparameters settings.

To find the optimal number of topics, topic coherence score is used. It is noticed that probabilistic models usually use perplexity to measure performance. For LDA, perplexity may not be the best to capture association between words. Mimno et al. (2011) gave a detailed review of the evaluation measures for topic models and proposed a new matric, the coherence score, to find high-quality topics that are also interpretable for human.

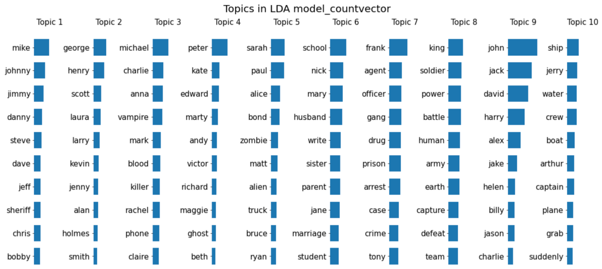


Fig 6: Topic distribution over words for count vector weighting scheme

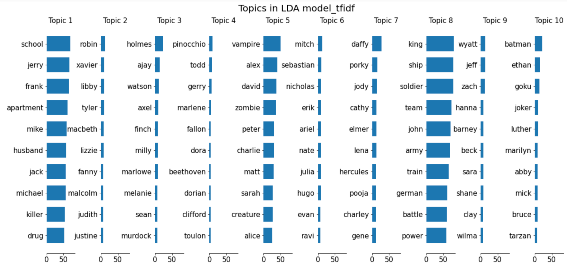


Fig 7: Topic distribution over words for *tf-idf* weighting scheme

Once the hyperparameters (number of topics, α and β) are all set, performance of the model is evaluated through comparing perplexity and topic distributions over documents. The comparison method is as follows. Each movie synopsis in the corpus could be viewed as a document. Each document has one or several tags assigned to it. Tags are treated as 'ground true' labels of documents. The tags assigned to each movie indicate the topics of the movie. We will focus on the top 5 tags in the corpus (as shown in Table I). These are movies with one single tag assigned to them.

TABLE I. TOP 5 TAGS IN THE MPST CORPUS, SERVE AS GROUND TRUE VALUE

Movie tags	**Number of movie synopses**
Murder	1004
Romantic	731
Violence	584
Psychedelic	437
Flashback	332

The next step is to construct a 'predicted' tag table from the result of the model. LDA will produce the distribution of documents over topics. Each document has a set of topics assigned to it and each topic is displayed with a probabilistic value, indicating the probability that a document is associated with that topic. We only consider the topic with the highest probabilistic value in a document. That topic is treated as the 'predicted' tag for that document.

The distribution of documents over topics will be transformed into a 'DataFrame' (a two-dimensional tabular table, as shown in Fig 8). The DataFrame has two columns. Column 'Topic ID' refers to the dominant topic in the corpus as assigned by the model. Column 'Num of documents' refers to the number of documents to which a dominant topic is assigned. This is treated as the 'predicted' table.

TABLE 2. M1 AND M2 STATISTICS

Model id	***M1***	***M2***
Stop words list	NLTK + spaCy (382 unique tokens)	NLTK + spaCy. (382 unique tokens)
Document frequency	min_df=0.001, max_df=0.02	min_df=0.001, max_df=0.02
Weighting techniques	Count vector	tf-idf
Doc-term matrix shape	14828 x 10893	14828 x 10893
Perplexity	3491	18386

Table I (ground true value) will be compared with predicted values (as illustrated in Fig.8). Specifically, the topic ID in Fig 8 will be checked against the keywords assigned to the corresponding topics by the model. These keywords (as illustrated in Fig.6) will then be compared with the movie tags in Table I. The number of documents in Fig 8 is checked against the number of movie synopses in Table I.

The comparison process depends on human interpretation and judgement. Therefore, the interpretability of the results of the topic model is important in this paper.

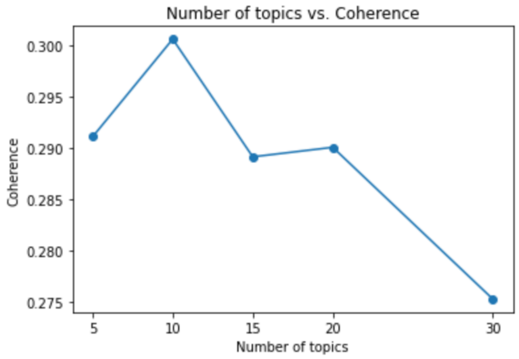


Fig 5: Coherence score and number of topics.

IV. RESULTS

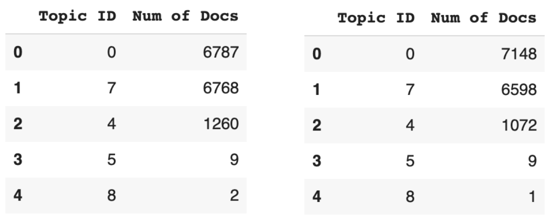
- *Experiment 1:*

To reduce feature space, it is important to limit the number of words. After applying the NLTK and spaCy stop word list and removing words that has less than 3 letters and words with more than 10 letters, we reduce the size of the dictionary to 70,266 tokens. The model is trained using Python Gensim's LDA model package to find the best coherence score. As shown by Fig 5, the optimal number of topics is 10.

After setting the model hyperparameters as fixed, I first experiment on stop word removals. The results of removing the most common words appear in all topics, such as 'find', 'kill', 'shoot', 'love' indeed improve the interpretability of the resulting topic inference. However, the effect is limited. After 2 addition the model stops to improve. It is also time consuming to eyeballing all the keywords and pick out the least useful ones.

TABLE 3. TOPIC DISTRIBUTIONS OF M1 AND M2

count vector			*tf-idf*		
M1			***M2***		
Topic 0	*Topic 7*	*Topic 4*	*Topic 0*	*Topic 7*	*Topic 4*
sheriff	king soldier human earth	zombie alien	school killer drug	king ship soldier german	zombie vampire



	Topic ID	Num of Docs
0	0	6787
1	7	6768
2	4	1260
3	5	9
4	8	2

	Topic ID	Num of Docs
0	0	7148
1	7	6598
2	4	1072
3	5	9
4	8	1

Fig 8: Document distributions over topic (left: *M1*, right: *M2*).

TABLE 4 . M3 AND M4 STATISTICS

Model id	***M3***	***M4***
Stop words list	NLTK + spaCy (382 unique tokens)	NLTK + spaCy. (382 unique tokens)

Document frequency	min_df=0.01,
	max_df=0.1
		min_df=0.01,
		max_df=0.1

Weighting techniques	Count vector	tf-idf
Doc-term matrix shape	14828 x 2672	14828 x 2672
Perplexity	1587	5435

When effect of removing stop words saturates, the next step is to focus on the adjustment of document frequency. It is more automatic and allows more time to explore the effect of context window size on model performance. For this reason, the remaining experiment focuses on adjusting document frequency.

By setting the minimum and maximum document frequency (min_df and max_df in Table III) for both weighting schemes (tf-idf as *M1* and traditional count vector as *M2*), the number of features is reduced to 10,893. It is interesting to note that perplexity scores on the training set of the two models are very different (see Table 2) while the document distributions over topics are similar (see Fig.8).

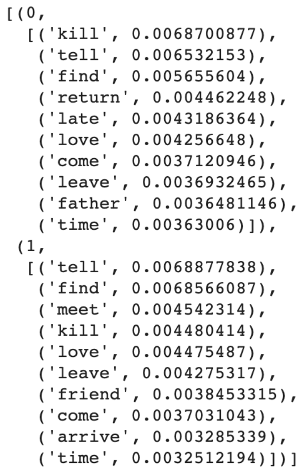
M1 and M2 has their document distribution over topics congregate around topic 0, topic 7 and topic 4 (Fig.8). Table 3 shows keywords that reveal interesting information about the top 3 topics. Topic 7 in both models are quite similar at the first glance as they are both associated with 'king' and 'soldier'. But M1topic 7 is more about space war and M2 topic 7 refers to wars on earth. Both models have similar distributions of keywords for topic 4. They differ at Topic 0. In general, both models capture the dominant tags in the movie synopses: violence and murder. Both models did not capture 'romantic' plots.

This may be because 70% of the corpus is either about murder or violence. The keywords inferred contain a lot of people's names which doesn't help interpretability (Fig.6 and Fig.7).

M1 and *M2* still have a lot of features, though they can capture the major topic of the corpus, the topics inferred lack granularity. *M3* and *M4* have significantly less features (2672) after adjusting document frequency (see Table 4). *M4* captures romantic, violence/murder, fantasy, sci-fi/space war, while *M3* captures violence and sci-fi/space war only. Notice that *M2* and *M4*, using tf-idf weighting scheme, have higher perplexity score than the models that use simple counting. *M2* and *M4* capture more of the granularity of the corpus and have more vivid words associated with topics. This seems to prove that perplexity as a measurement for LDA performance is not the optimal option if the purpose is to have more interpretable topics. However, the perplexity score of models using the same weighting schemes decreased as the model topic interpretability increases (*M3* is lower than *M1* and *M4* is lower than *M2*, see Table 3 and Table 2).

- *Experiment 2:*

Following the conclusion of experiment 1, that the tf-idf captures the semantic details of a corpus better than simple document term frequency counting, the next experiment will be carried out on a smaller corpus, using only the movie synopses with only one tag: 'murder' (1007 movies) or 'romantic' (731 movies). The reason is that they are two distinct themes, easy to differentiate for human interpretation. They are the dominant tags in the corpus, providing a decent number of documents/synopses for LDA.



```
[(0,
  [('kill', 0.0068700877),
   ('tell', 0.006532153),
   ('find', 0.005655604),
   ('return', 0.004462248),
   ('late', 0.0043186364),
   ('love', 0.004256648),
   ('come', 0.0037120946),
   ('leave', 0.0036932465),
   ('father', 0.0036481146),
   ('time', 0.00363006)]),
 (1,
  [('tell', 0.0068877838),
   ('find', 0.0068566087),
   ('meet', 0.004542314),
   ('kill', 0.004480414),
   ('love', 0.004475487),
   ('leave', 0.004275317),
   ('friend', 0.0038453315),
   ('come', 0.0037031043),
   ('arrive', 0.003285339),
   ('time', 0.0032512194)])]
```

Fig 9: Topic keywords produced by LDA (alpha=0.8, eta=0.001, simple count vector), trained on 'murder' and 'romantic' movie synopses. 'find', 'tell', 'kill', 'love' are the most frequent words appear in the topic words distributions under various tunings of model hyperparameters. Both (alpha=0.8, eta=0.001) and (alpha=0.1, eta=0.4) produces the highest coherence score when the topic number is 2. However, these sets of hyperparameter does not yield better topic inference than the default setting of Gensim and Scikit Learn LDA default setting.

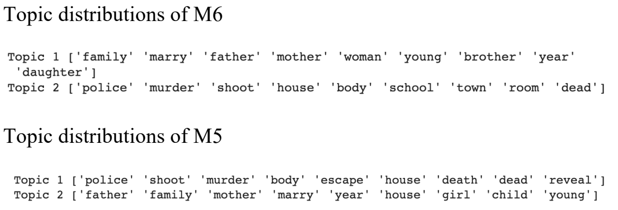
The 'murder' and 'romantic' dataset (1738 documents in total) is pre-processed in the same way as *M3* and *M4* but with additional stop words removed. The lists of stop words are obtained while tuning the hyperparameters of LDA (see Secion 3 D), when ‘murder’ and ‘romantic’ dataset is trained (see Fig 9).

TABLE 5 . M5 AND M6 PERPLEXITY SCORE

	Document frequency	*M5(count)*	*M6(tf-idf)*
1	min_df=0.01, max_df=0.1	1648	3208
2	min_df=0.1, max_df=0.4	232	327

Combining document frequency adjustment with the addition of these common words into the stop words list, the resulting quality of topic inference is significantly improved (see Fig. 10).

The second configuration of document frequency in Table 5 yields better topic inference for M5 and M6 which can be distinguished clearly as ‘murder’ and ‘romantic’ based on the keywords (Fig. 10).



Topic distributions of M6

```
Topic 1 ['family' 'marry' 'father' 'mother' 'woman' 'young' 'brother' 'year'
 'daughter']
Topic 2 ['police' 'murder' 'shoot' 'house' 'body' 'school' 'town' 'room' 'dead']
```

Topic distributions of M5

```
Topic 1 ['police' 'shoot' 'murder' 'body' 'escape' 'house' 'death' 'dead' 'reveal']
Topic 2 ['father' 'family' 'mother' 'marry' 'year' 'house' 'girl' 'child' 'young']
```

Fig 10: Topic distributions of M5 and M6. The topic number is configured to 2 for M5 and M6 as a result of comparing coherence scores obtained by iterating the LDA through several topic numbers.

- *Experiment 3:*

As LDA performs very well in capturing the main themes of the corpus, the ‘murder’ synopses are extracted as a dataset to drill down for more details. It is split into 80% training set and 20% held-out test set. The document frequency controls the feature space. As the number of features decrease, the perplexity of the model is lower for both the training set and test set. Models using simple count vectorizer to transform the document and words into doc-term matrix are less sensitive to the number of tokens in terms of perplexity score, while models using tf-idf weighting technique react strongly to the change of token numbers.

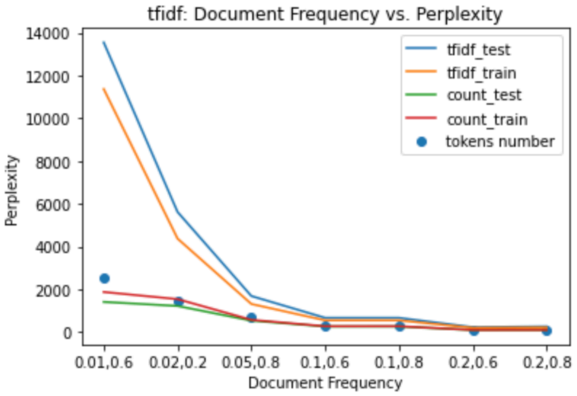


Fig 11: Perplexity score decreases as the number of feature tokens decrease. For models trained on ‘murder’ synopses only. The curves of the tf-idf and count vector models elbow at (0.05, 0.8), referring to the minimum document frequency and maximum document frequency respectively.

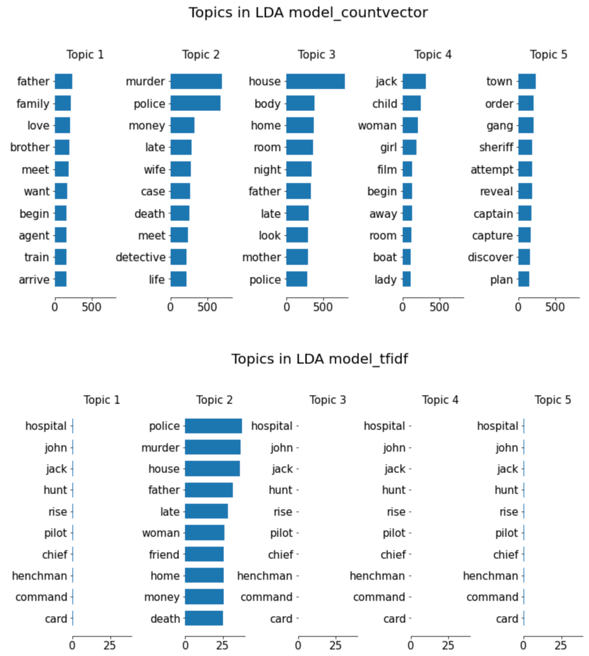


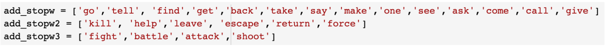
Fig 11. Topic keywords inferred by models using tf-idf and count vector weighting scheme, minimum doc-frequency is 0.05, maximum doc-frequency is 0.8, respectively trained on 80% of movie synopses tagged ‘murder’. The tf-idf LDA model has repetitive words in 4 of the 5 topics.

The keywords assigned to topics (see Fig 12) using the (0.05, 0.8) document frequency setting, revealed no more information than the inference of models trained on the whole dataset (see Fig 6 and Fig 7). The tf-idf LDA model yields poor quality topic inference under this frequency setting.

V. CONCLUSION

This paper set out to explore three pre-processing techniques: n-grams, stop word removal, and document frequency adjustment. The one technique that make the most difference for the model results is document frequency adjustment. By removing the lowest frequency and highest frequency words, the models produce more meaningful results (Fig 6 and Fig7) than using the whole corpus. But the benefit of document frequency adjustment is more obvious for large dataset with known labels.

Stop word removal technique is explored when the hyperparameters are tuned, revealing a set of common words that repetitively appear across different settings. As shown in Fig. 13, the list is added post training iteratively. The words in the add_stopw list appear repetitively in topic inference during the experimental tuning of alpha and eta. The words are collected and added to the stop words list in the pre-processing step and are removed in the next training. This process is repeated 3 times. After that, the model topic inference shows more variation as the hyperparameters are changed. The



```
add_stopw = ['go','tell', 'find','get','back','take','say','make','one','see','ask','come','call','give']
add_stopw2 = ['kill', 'help','leave', 'escape','return','force']
add_stopw3 = ['fight','battle','attack','shoot']
```

Fig 13: Stop words list added post training. See also Fig 9.

quality of topic inference is improved when the additional stop words lists are added to a smaller dataset with 2 clear themes (*M5 and M6*).

The evaluation for the models during experiment rely heavily on human interpretation and judgement. Two statistics are used for evaluation. First, coherence score is used as a measurement to decide the number of topics. Second, perplexity scores of the LDA models are compared for feature selection. However, as illustrated by the keywords inferred by the models using different term weights and features, the perplexity score does not reflect the quality of the LDA models, especially the interpretability of the keywords. The coherence score is sensitive to the hyperparameters of LDA, making it hard to decide whether the number of topics chosen is really the optimal one for the model and dataset. For this dataset with clear tags, the topics are known. This renders human judgement less biased because it is based on the corpus. As shown in the smaller dataset (i.e. movie synopses with only 'murder' or 'romantic' tag), the number of topic is set to 2 not only because the coherence score tells so, but also because of the prior knowledge that the dataset indeed has 2 very different themes. The keywords inferred, as a result, are more meaningful.

However, as the dataset gets smaller (i.t. movie synopses with only 'murder' tag) and the theme too general, it is hard to generate satisfying keywords. The models experimented cannot learn the topics very well. This in a way reflects the challenge of LDA. As it only takes the words and documents as input, it reflects the corpus as just as it could. However, during implementation, a lot of tuning is involved. As the experiment goes on, I realise I am trying to look for words I want to see. If, to my knowledge, the words allocated are not meaningful enough, I go back and adjust the pre-processing techniques. This introduces bias and should be aware of.

This paper concludes that tf-idf is a good option for exploring more granular topics for movie synopses applied on LDA. To capture the general themes of a corpus, simple count vector is sufficient. Adjusting document frequency yields better topic inference. And finally, compiling specific stop words lists improves the quality of topic inference for LDA.

Future work that focuses on topic modelling on movie synopses could investigate the removal of person names as stop words. As names dominate topics whatever the document frequency is.

References

Albalawi, R., Yeap, T.H. and Benyoucef, M. (2020). Using Topic Modeling Methods for Short-Text Data: A Comparative Analysis. Frontiers in Artificial Intelligence, 3.

Blei, D., Edu, B., Ng, A., Jordan, M. and Edu, J. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, [online] 3, pp.993–1022. Available at: https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf.

Calvo - González, O., Eizmendi, A. and Reyes, G. (2018). Winners Never Quit, Quitters Never Grow: Using Text Mining to Measure Policy Volatility and its Link with Long-Term Growth in Latin America. [online] papers.ssrn.com. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3105770 [Accessed 21 Nov. 2021].

David M Blei, Andrew Y Ng, and Michael I Jordan. (2003). Latent Dirichlet allocation. The Journal of Machine Learning Research 3:993–1022.

Denny, M. J. and Spirling, A. (2018) "Text Preprocessing For Unsupervised Learning: Why It Matters, When It Misleads, And What To Do About It," *Political Analysis*. Cambridge University Press, 26, pp. 168–189. doi: 10.1017/pan.2017.44.

Griffiths, T.L. and Steyvers, M. (2004). Finding scientific topics. Proceedings of the National Academy of Sciences, 101(Supplement 1), pp.5228–5235.

Kar, S., Maharjan, S., López-Monroy, A.P. and Solorio, T. (2018). MPST: A Corpus of Movie Plot Synopses with Tags. [online] ACLWeb. Available at: https://aclanthology.org/L18-1274/ [Accessed 16 Nov. 2021].

Mcauliffe, J. and Blei, D. (2008). Supervised Topic Models. [online] Neural Information Processing Systems. Available at: https://papers.nips.cc/paper/2007/hash/d56b9fc4b0f1be8871f5e1c40c0067e7-Abstract.html [Accessed 18 Nov. 2021].

Mimno, D., Wallach, H., Talley, E., Leenders, M. and Mccallum, A. (2011). Optimizing Semantic Coherence in Topic Models. [online] Association for Computational Linguistics, pp.262–272. Available at: http://dirichlet.net/pdf/mimno11optimizing.pdf [Accessed 19 Nov. 2021].

Schofield, A. and Mimno, D. (2016). Comparing Apples to Apple: The Effects of Stemmers on Topic Models. Transactions of the Association for Computational Linguistics, 4, pp.287–300.

Schofield, A., Magnusson, M. and Mimno, D. (2017). Pulling Out the Stops: Rethinking Stopword Removal for Topic Models. [online] ACLWeb. Available at: https://aclanthology.org/E17-2069/ [Accessed 16 Nov. 2021].

Schofield, A., Magnusson, M., Thompson, L. and Mimno, D. (2017). Pre-Processing for Latent Dirichlet Allocation. [online] www.semanticscholar.org. Available at: https://www.semanticscholar.org/paper/Pre-Processing-for-Latent-Dirichlet-Allocation-Schofield-Magnusson/937345c3bb4a59010ffa17f8bd7db456e2e2d048 [Accessed 16 Nov. 2021].

Serrano, Luis. (2020). Latent Dirichlet Allocation (Part 1 of 2). [online] https://www.youtube.com/watch?v=T05t-SqKArY&t=628s.