# CS 341-NL Final Project Report

Ranjani Ramanathan, Jenna Hammond

TOTAL POINTS

## 90 / 100

QUESTION 1

### 1 Report (select all pages) 90 / 100

+ **0 pts** Correct

+ **90 Point adjustment**



gradescope

Ranjani Ramanathan, Jenna Hammond
COMSC-341-NL
13 December 2019

# Creating a Movie Recommendation System with Topic Modeling

## Abstract

Our research project seeks to detect similarity between movies and generate movie recommendations using a content-based approach. We used topic modeling, specifically Latent Dirichlet Allocation, to design a recommendation system that uses the prevalent language and themes of movies to compute similarity based on their scripts. We present two methods of selecting similar movies, one using the dominant topics of a film, and the other using the full distribution of topics. To evaluate our approaches, we performed a user study in which the user was asked to rate sets of recommendations from each approach, as well as our baseline approach, which uses document similarity of scripts as a whole, with no consideration of topics. Ultimately, our results showed that using the full distribution of topics is the most effective method, but we found that our overall scores were lower than desired, and performance was fairly inconsistent as a result both of the nature of screenplays and our approach to topic modeling.

## 1. Introduction

With the growth of streaming services such as Netflix, automating movie recommendations has become a very popular problem in computer science. This is frequently done using large amounts of user data from the service in question or online movie review websites. However, in the absence of such data, it could be quite useful to determine similarity based solely on the content of a film. In order to tackle such a problem, one must be able to detect content themes within documents. Fortunately, topic modeling is intended to do just that, using unsupervised learning find clusters of words that share a semantic domain, in hopes to discover prevalent subjects or themes within a document. Thus, our project aims to generate movie recommendations using topic modeling. We developed a topic model using Latent Dirichlet Allocation on screenplays of various films, and created two separate methods for topic-based recommendations of movies within our corpus.

### 1.1 Related Work

Topic modeling is a quickly growing area of NLP research. Though it is often used in data analysis, such as examining what a piece of text can reveal about the author, there are plenty of researchers finding new applications for this field of study (Schwart et al., 2013). Even within the domain of film, there is a significant amount of work being done on how to apply topic modeling in various ways, though we are focusing on the problem of movie recommendations. A common approach to recommendations, even outside of topic modeling research, uses user data and feedback to find films that tend to appeal to similar groups of people (Diao, et al., 2014).

However, as mentioned, our project aims to build a system in the absence of user data, taking a content-based approach. In 2016, a group of researchers built such a system using topic modeling of the subtitles 160 films, with fairly positive results (Bougiatiotis & Giannakopoulos). Our research is quite similar, though our model is simpler. Another important difference is our usage of full scripts rather than subtitles, based on the idea that more information about a film is provided by the screenplay rather than the dialog alone. This decision also had a precedent in the successful creation of an automated movie genre classifier which used LDA topic modeling of scripts (Chao & Sirmorya, 2016).

## 2. Methodology

### 2.1 Latent Dirichlet Allocation (LDA)

We built our model with the probabilistic Latent Dirichlet Allocation algorithm for topic modeling, as used by Chao and Simorya (2016). This algorithm takes 3 parameters: *K,* the number of topics in the model; $\alpha$, a constant that affects the level of mixture of the topics within a document; and $\beta$, a constant that affects the number of words in a single topic. It begins by assigning a random topic to each word within the documents, and then reiterates many times, improving the model by selecting the distribution of topics with the highest probability, using the formula $P(topic) = P(topic|document) * P(word|topic)$. It computes this for every topic for each word in the corpus, and re-assigns each word to the topic with the highest probability. Eventually, topics are assigned more consistently to clusters of words, which are considered to be related to one another in some way within the model. Ultimately, every document is represented as a distribution of these topics based on the word assignment.

Our model uses the implementation of LDA provided by MALLET, a machine learning toolkit created by University of Massachusetts Amherst.[1] We used 10 topics in our model, and alpha and beta values of 50 and .01. These were selected by testing different values and judging the coherency of the topics produced by the model.

### 2.2 Data

We used the Film Corpus 2.0 dataset, compiled by University of California Santa Cruz.[2] It consists of 1,068 full screenplays from popular films in text files that are tagged by genre. One of the immediate challenges we faced when working with this data was the frequency of character names of directions. As all documents are in raw script format, each line in the movie is preceded by the name of the speaker. Similarly, there are certain words that are standard in screenplays and used in scene descriptions, such as cont, int, and ext, which also appear a disproportionate amount. With this prominence, these words severely skewed initial models and rendered our topics incoherent. In order to clean the data, we utilized an NLP library called spaCy[3] for its named entity recognition capabilities to remove all names that it could detect from

[1] http://mallet.cs.umass.edu/
[2] https://nlds.soe.ucsc.edu/fc2
[3] https://spacy.io/

the scripts themselves. This still did not account for all names, so we created a custom list of stopwords to cover most of the remaining names, fictional or otherwise, as well as common scene descriptors. Most of the names were from the Names Corpus[4], a dataset containing more than 8,000 human names, and the rest were added based on observation. This was sufficient to improve topic coherency, so that our model produced topics such as those shown in Figure 1 below.


Figure 1: Word clouds of the most common words of two topics from the model

**2.3 Recommendations**

In developing our recommendation systems, we decided to try two different approaches to finding similar movies based on our model. In topic modeling, documents are considered to be combinations of the model's topics with varying ratios, so each document can be represented as a vector of $K$ percentages, one for each topic $t$, as shown in the equation below.

$$V[t] = \frac{\#\ of\ words\ in\ topic\ t}{\#\ of\ words\ in\ document}$$

Each of our approaches used this distribution information to generate recommendations.

The first method, referred to as the "top two topics" approach, relied only on the two most prominent topics of the scripts to detect similarity. That is, the distribution values for the two topics which had the highest percentages for the input film. It begins by selecting all movies with the same two most common topics from the corpus. From that list, it chooses the three movies with the closest value to the input film's percentage for the first topic, and then two movies with the closest value for the second most prominent topic. This method outputs movies with strong associations of first and second topic, the underlying idea being that this would demonstrate the importance, or lack thereof, of dominant topics in assessing similarity.

The second method, referred to as the "topic vector" approach, computed similarity using the entire topic distribution vector. For this approach, the cosine similarity of the topic distribution vectors was calculated for the input movie and every other film in the corpus, and the five movies with the highest cosine similarity were selected for the recommendations. Cosine similarity, shown in the formula below, is used to measure the closeness of two vectors, so it seemed to be an appropriate choice for comparing these distributions, which are easily

[4] https://www.kaggle.com/nltkdata/names

represented as vectors. The key difference in this method is that accounts for all topics, not just the more dominant ones, as less prominent ones may still provide a lot of information about the content of a movie. Despite these differences, there was still regular overlap between the recommendations of these approaches, as shown in Figure 2.


Figure 2: The set of recommendations given by each approach for the 1982 science fiction thriller *The Thing*. Both provide various high intensity action/thriller films, though they only share one movie.

## 3. Evaluation

In order to evaluate our approaches, we performed a user study with 20 people. Each user was asked to select a movie from the corpus and received three sets of five recommendations for their chosen film. Each set was generated by a different approach: one using the "top two topics" approach, one using the "topic vector" approach, and one using the baseline approach to which we compare our own recommendation system. In the baseline approach, recommendations are chosen based on word embeddings of the movies' scripts as a whole, calculating the document similarity of two scripts using the word vectors included in spaCy's English language model. The five films with the highest similarity values were chosen. This seemed like an appropriate comparison as a straightforward approach to using movie scripts for recommendations. After

receiving the recommendations, the users rated each set on a scale from 1 to 10 based on the movies' perceived relevance to the selected film. The average scores, highest scores, and rankings of the recommendation systems are shown in Figure 3 below.

| Approach | Average Rating | Highest Rating | # of Times Scored Highest |
|---|---|---|---|
| Baseline | 4.842 | 9 | 6 |
| Top Two Topics | 4.578 | 7 | 3 |
| Topic Vector | 5.763 | 8 | 11 |

Figure 3: The average 1-10 rating for each approach, the highest score received, as well as the number of times they received the highest score of the three sets.

## 4. Discussion

Overall, we received mixed results from our evaluation. The topic vector approach did perform above the baseline in average rating and in user rankings, but the difference in average rating (.921 points) was fairly small. The baseline received the highest score of all user ratings, and as a whole, the average scores were fairly low. The top two topics approach performed the worst in all three assessment categories, which tells us that only using the two most prominent topics is not sufficient to detect similarity. Users almost always gave the topic vector recommendations a higher rating, implying that using the full topic distribution is better for detecting similarity, despite the overlap which did occur regularly. Thus, we can conclude that information regarding topics that are not prevalent in a film are also useful in qualitative assessment based on their scripts.

Despite the relative consistency of the topic vector approach scoring higher, the system as a whole was inconsistent in its performance, which is confirmed by the middling average scores. When a person examines the qualities of a movie, they tend to see themes, tropes, storylines, and characterizations. However, topic modeling only has the ability to find word patterns, not interpret the movie. Thus, our recommendations were often selected based on similarities in settings, speech patterns, and recurring objects. Movies could be grouped together due to similar environments or the vernacular of central characters. While this was sometimes sufficient to find relevant films, it occasionally resulted in drastically different films being recommended. For example, *Wall-E* is a light-hearted children's movie, but its recommendations within all approaches were largely science fiction horror/thriller movies, as *Wall-E* takes place in a futuristic, somewhat post-apocalyptic world. As a result of this tendency, the recommendation system tended to perform the best when given a genre film, such as a science fiction or horror movie. This may be due in part to the presence of a significant amount of information regarding setting within a screenplay. While this additional information was thought to be a strength of our

approach, it is worth noting that the dialog-based approach used by Bougiatiotis and Giannakopoulos did not encounter this problem.

### 4.1 Limitations

Ultimately, we believe that some of the weaknesses of our recommendation system are a result of our model itself. Due to time constraints, we were unable to spend as much time refining it as we would like. In addition to the ability to spend time adjusting it based on our own judgment, there are concrete metrics that are often used to evaluate topic models. Topic coherence, for example, can be computed in such a way that has proven consistent with human judgment, and intra-inter topic distance calculates the "semantic tightness" of the documents, assessing the consistency of topic distribution throughout a document and the relative uniqueness of each document's distributions. These are used in topic modeling research to optimize and evaluate a model, and may have proven quite useful in our research as well (Bougiatiotis & Giannakopoulos, 2016). Besides the time constraints, our approach itself may have been hindered by its simplicity. Topic modeling research often takes a hybrid approach to building the model, combining other natural language processing techniques with their chosen topic modeling algorithm. As topic modeling and LDA were already new to us, we lacked the expertise required to implement these more complex models, which may have also been detrimental to the efficacy of our recommendation system.

References

Bougiatiotis, K., & Giannakopoulos, T. (2016). Content Representation and Similarity of Movies based on Topic Extraction from Subtitles. In *Proceedings of the 9th Hellenic Conference on Artificial Intelligence* (pp. 17:1-17:7). New York, NY: ACM. doi: 10.1145/2903220.2903235

Chao, B., & Sirmorya, A. (2016). Automated Movie Genre Classification with LDA-based Topic Modeling. *International Journal of Computer Applications 145*(13).

Diao, Q., Qiu, M., Wu, C., Smola, A.J., Jiang, J., and Wang, C. (2014). Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 193-202). New York, NY: ACM. doi: https://doi.org/10.1145/2623330.2623758

Schwart, H.A., Eichstaedt, J.C., Kern, M.L., Dziurzynski, L., Ramones, S.M., Agrawal, M., et al. (2013). Personality, Gender, and Age in the Language of Social Media: The Open-Vocabulary Approach. *PLoS ONE 8*(9): e73791. doi: 10.1371/journal.pone.0073791

1 Report (select all pages) **90 / 100**

+ **0 pts** Correct

+ **90** **Point adjustment**



gradescope