Natural Language Processing

Assignment 3

Project Nature and Content

Julianna Antonchuk, Yabo Gao

MSDS 453, Northwestern University

29 August 2020

Contents

Abstract

In the final research assignment, we set an objective to compare representations of human content perception with algorithmic methods such as clustering, classification, and sentiment analyses. For the final research assignment, we leverage the course-wide corpus with a heterogeneous set of movie reviews. With an ontology and category developments we abstract human perception of a given corpus to further compare the results of human topic modeling to algorithmic partitioning. In our research, we focus our problem scope around adventure and action movie genres representations. We analyze how well these categories are represented in the formed clusters, classification, and dense neural network outputs. For clustering learning, we apply the k-Means clustering method. In our assessment of the clustering results, we aim to identify the appropriateness and adequacy of the cluster assignments, compare how the genres of interest are represented in the formed clusters, and observe the algorithmic associations. For the classification method, we employ Random Forest Classifier to observe algorithmic groupings of the evaluated corpus. We take the corpus-wide reference term vector generated by the term frequency-inverse document frequency (tf-idf) and the word-embedding (Doc2Vec) methods as an input to these learning methods. For the sentiment analysis, we develop a simple Multilayer Perceptron (MLP) model that classifies encoded documents as those that either belongs to action and adventure genres or not. We apply four different methods of tokenization for scoring words – binary, count, tf-idf, and frequency and choose a configuration with the highest accuracy distribution. With the performance assessment of algorithmic text representation learning methods, we observed how k-Means clustering, Random Forest Classification, and a simple DNN demonstrated very promising results that are closely aligned with human judgment and hypothesis with the regards to a genre of interest.

Keywords: *natural language processing, text analysis, text categorization, text representation, topic modeling, text clustering analysis, k-means clustering, text classification analysis, random forest classification, deep learning, multilayer perceptron, dense neural network, sentiment analysis, bag-of-words vectors, reference term vector, term frequency-inverse document frequency, tokenizer, word embeddings, Word2Vec, Doc2Vec, equivalence classes.*

1 Introduction

1.1 Background

We conduct the research to observe how well algorithmic methods perform text representation learning and whether their results demonstrate some relative closeness to a given content perception by humans. The course-wide corpus is represented with 64 documents, content of which takes on heterogenous movie reviews. We develop an ontology and categorical mapping to build a semantic abstraction of the corpus and create a higher-order genre classification to further compare human-defined representations with the algorithmic outputs. In the research assignment, we focus on how distinctly a particular movie genre is represented in the outputs of the evaluated text representation learning methods. The algorithmic methods considered in the research are k-Means clustering, Random Forest Classification, and simple DNN models. For the data preprocessing step, we create the corpus-wide reference term vector produced by the term frequency-inverse document frequency (tf-idf) and the word-embedding (Doc2Vec) methods. In the sentiment analysis, we consider four different comparable methods of tokenization for scoring words – binary, count, tf-idf, and frequency. We aim to evaluate similarity and accuracy among the algorithmic methods and assess how well their results align with the human understanding of the context semantics.

1.2 Objectives

The key objective is to compare text representations perceived by humans with algorithmic representations generated by means of clustering, classification, and sentiment analyses.

1.3 Approach

We approach the research design sequentially:

1. With a categorical development, we create higher-order semantic representations. The research focuses on movie genres as high-level categories for the evaluated corpus.
2. With ontology development, we aim to abstract semantic structure, respective perception, and understanding of the evaluated corpus by humans.
3. We focus on the categories of the action and adventure movie genres in our research. We seek to learn how these genres are represented in the corpus algorithmically.
4. In the clustering analysis, we evaluate how clusters are formed with respect to the evaluated genre category. We consider the term frequency-inverse document frequency (tf-idf) and the word-embedding (Doc2Vec) methods.
5. In the classification analysis, we assess how the corpus is partitioned with respect to the genre of interest. We apply similar preprocessing vectorization methods as above.
6. With the sentiment analysis, we apply a deep learning model to predict sentiment classification in respect to the genre association.

2 Methodology

2.1 Research Questions

The research pursues to answer the following aspects:

1) Develop ontology to obtain semantic structure and knowledge of the corpus.
2) Identify high-level categorical associations (i.e. genre-based) in the evaluated corpus.
3) Assess the performance of algorithmic text representation learning methods.
4) Compare algorithmic methods performance with regards to a genre of interest.
5) Evaluate how text representations perceived by humans aligned with algorithmic representations generated by means of clustering, classification, and sentiment analyses.
6) Identify an optimal word scoring method for the DNN model.
7) Recommend an algorithmic approach for the evaluated corpus.

2.2 Research Design

We design the research experiment to meet the objective of the problem statement – alignment of algorithmic text representation with human context perception and understanding. Therefore, we develop ontology and categorical mappings of the corpus and reference them as human-generated representations. In our research, we aim to evaluate how closely text representations generated by humans aligned with algorithmic text representation learnings. We focus on a particular movie genre – adventure and action and seek to observe how it is represented in the formed clusters, classification groupings, and neural network predictions. We evaluate cluster formations and classification groupings based on the input matrix (tf-idf vs. Doc2Vec) as well as four different methods of tokenization for the bag-of-words model in our neural network architecture.

2.3 Instruments

For our research development, we apply methods of feature extraction for text, clustering, label encoding, model evaluation and selection, tree-based models of the scikit-learn machine learning library. We leverage methods of the genism library for the text vectorization and word embedding along with Keras API for the development of the neural network.

2.4 Data Collection

Across two sections of the MSDS-453 Natural Language Processing summer 2020 course at Northwestern University, we have gathered 64 documents with movie reviews. The corpus takes on at least two documents from each student containing either two reviews on the same movie selected by a student, or two reviews on two different movies selected by a student. The theme of the corpus is a movie review. Reviews and movies are heterogeneous and disperse in their nature, making a text processing task challenging and interesting.

3 Research Results

3.1 Categorical Mapping Development: Results

For the research assignment, we focus on understanding how a genre of action and adventure is represented in the course-wide corpus. Therefore, we create a genre-based categorical mapping for the movie plots captured in the corpus. We distinct the movies that fall into a category of action and adventure from those that do not. The categorical mapping is consulted with the Internet Movie Database (IMDb) owned by Amazon. The online database contains information related to films, television programs, home videos, video games, and streaming content online. The database has approximately 6.5 million titles and 10.4 million personalities. The contribution to the database is crowd-sourced ("IMDb", Wikipedia).

Below, we present a table where we alphabetically list all the movies (*n=37*) from the course-wide corpus and their associated genre from IMDb. We assign a label of ***"1"***, if a movie belongs to either action and/or adventure genre, and ***"0"***, if a movie does not belong to either of these genres, i.e. belongs to another genre, i.e. "others".

Table 1. Genre-based Categorical Mapping. Label Assignment.

Movie Title	IMDb Genre	Label
1917	Drama, War	0
50 First Dates	Comedy, Drama, Romance	0
A.I.: Artificial Intelligence	Drama, Sci-Fi	0
Anchorman	Comedy	0
Artemis Fowl	Adventure, Family, Fantasy	1
Bad News Bears	Comedy, Sports	0

Big Daddy	Comedy, Drama	0
Blade Runner	Action, Sci-Fi, Thriller	1
Blade Runner 2049	Action, Drama, Mystery	1
Coco	Animation, Adventure, Family	1
Dumb and Dumber	Comedy	0
Eternal Sunshine of the Spotless Mind	Drama, Romance, Sci-Fi	0
Finding Nemo	Animation, Adventure, Comedy	1
Frozen	Animation, Adventure, Comedy	1
Frozen 2	Animation, Adventure, Comedy	1
Garden State	Comedy, Drama, Romance	0
Get out	Horror, Mystery, Thriller	0
How to Train a Dragon	Animation, Action, Adventure	1
Inception	Action, Adventure, Sci-Fi	1
Just go with it	Comedy, Romance	0
Knives Out	Drama, Comedy, Crime	0
Lord of the Rings	Action, Adventure, Drama	1
Moana	Animation, Adventure, Family	1

Monsters University	Animation, Adventure, Comedy	1
Parasite	Crime, Drama, Thriller	0
Reservoir Days	Crime, Drama, Thriller	0
Secret Life of Pets	Animation, Adventure, Comedy	1
Terminator 2: Judgment Day	Action, Sci-Fi	1
The Farewell	Biography, Comedy, Drama	0
The Hateful Eight	Crime, Drama, Mystery	0
The Hobbit	Animation, Adventure, Family	1
The Imitation Game	Biography, Drama, Thriller	0
The Lego Movie	Animation, Action, Adventure	1
The Lion King	Animation, Adventure, Drama	1
The Predator	Action, Adventure, Sci-Fi	1
Tommy Boy	Adventure, Comedy	1

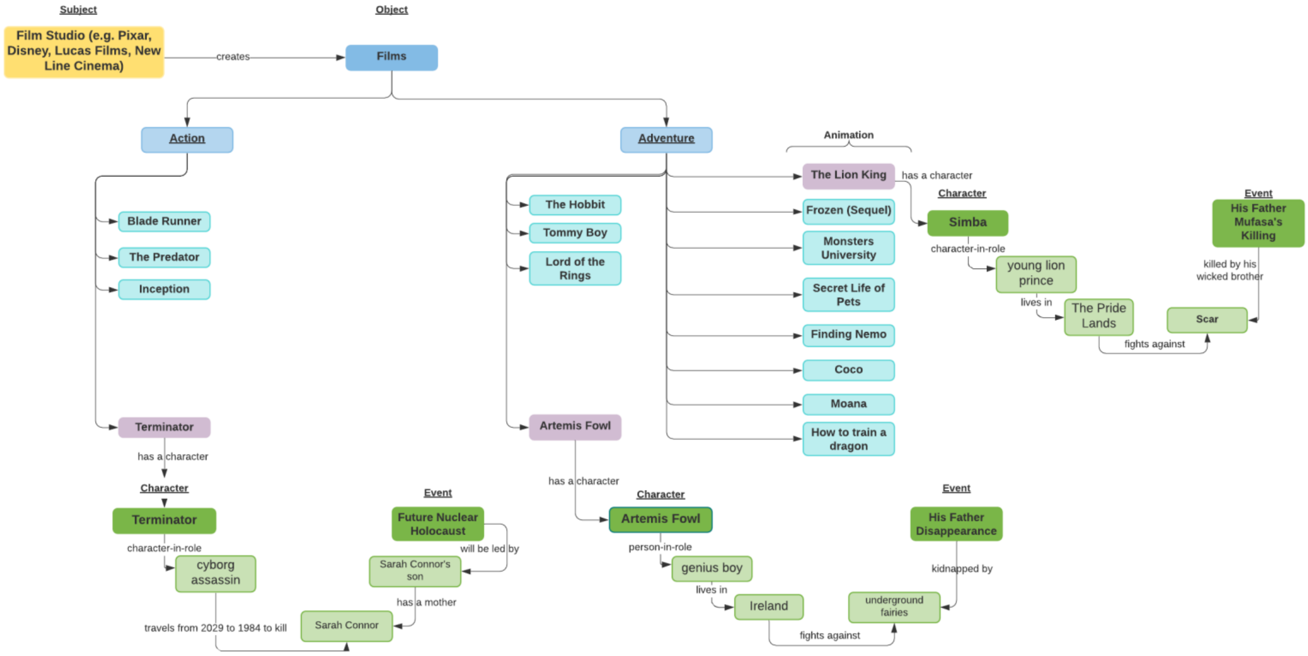
With the newly added labeled data, we generate the term frequency-inverse document frequency (tf-idf) and the word-embedding (Doc2Vec) matrices for the clustering, classification, and sentiment analyses tasks.

3.2 Ontology Development: Results

When developing an ontology, we focused on how action and adventure genres are represented in the corpus. Adventure genre includes animation films, therefore we observe such film studios as Pixar and Disney as acting subjects. Generally, adventure movies take on storylines with characters who are set on a quest to overcome some challenges, conquer an evil side, develop themselves, save their beloved ones or the world. We evaluated the entities and relationships for such representative movies of this genre as “The Lion King” and “Artemis Fowl”. Their ontologies are color-coded in green.

The action genre is characterized by special visual effects and computer-generated imagery (CGI). Their storylines take on scientific, futuristic, philosophical, very often destructive, and violent scenes. Typically, a leading character is involved in a series of events that include extended fighting, physical feats, and frantic chases, struggles against incredible odds. For the action movie genre, we explored “Terminator”.

Table 2. Action and Adventure Ontology.



3.3 Clustering Analysis. K-Means Clustering: Results

In this section, we present the analysis and discussion around the results of the clustering exercise applied on the matrix generated with the term frequency-inverse document frequency (tf-idf) vectorizer. The term frequency-inverse document frequency (tf-idf) vectorizer transforms the text into a vector of numbers allowing further algorithmic processing of the text. We explore scenarios where we partition the vectorized data into k clusters (*k=2*). Each observation in data gets assigned to a cluster with the nearest mean, i.e. a cluster center or a centroid. This results in a partitioning of the data space into Voronoi cells. As a method, k-Means clustering aims to minimize within-cluster variances, optimizing squared errors, i.e. squared Euclidean distances. If a problem aims to minimize regular Euclidean distances, k-Medians and k-Medoids can be used to address optimizing within-cluster variances. K-Means clustering tends to find clusters of comparable spatial extent.

In a set of observations $(x_{1,}\, x_{2,}\, \dots, x_n)$, where each observation is a d-dimensional real vector, k-Means clustering aims to partition the *n* observations into *k ($\leq$ n)* sets S = $\{S_{1,}\, S_2, \dots, S_k\}$, so as to minimize the within-cluster sum of squares. The objective function is as follows:

$$\arg min_s \; \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 \; = \arg min_s \; \sum_{i=1}^{k} |S_i| \, Var \, S_i,$$

where μ_i is the mean of points in S_i. It is equivalent to minimizing the pairwise squared deviations of points in the same cluster:

$$\arg min_s \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{x,y \,\in\, S_i} \|x - y\|^2$$

The equivalence can be deduced from identity:

$$\sum_{x \in S_i} \|x - \mu_i\|^2 = \sum_{x \neq y \in S_i} (x - \mu_i)(\mu_i - y)$$

Since the total variance is constant, this is equivalent to maximizing the sum of squared deviations between points in different clusters, i.e. between-cluster sum of squares, which follows from the law of total variance ("*k*-Means clustering", Wikipedia.).

We apply an unsupervised clustering algorithm to iteratively differentiate groupings with dominant words in each group. We evaluate how the terms that formed clusters well represent the respective groupings, and whether there is a semantical and contextual alignment with ground-truth understanding. Specifically, we evaluate partitions that discriminate clusters per genre-based label, i.e. "Action/Adventure" vs. "Others". We compare the clustering output produced based on the tf-idf matrix that is unaware of the label to the results that consider an imposed categorization. Below, we share the results of two iterations where we computed similarity ratios between clustering outputs that (i) considered the introduced categorization and (ii) that did not.

Table 3. k-Means Clustering. Similarity Ratio. TF-IDF vs. TF-IDF Custom.

Iteration	Action/Adventure	Others
1	0.464	0.535
2	0.526	0.474

We observe relatively promising results. They suggest that k-Means clustering analysis (*k=2*) performed on the tf-idf matrix without an introduced feature demonstrated a highly aligned output to the one generated on the custom matrix with a genre category feature. Here, we observe a congruency between algorithmic output and human judgment.

In the research, we would like to employ word embedding methods such as Doc2Vec. Doc2Vec and Word2Vec methods were developed by Mikilov and Le (2014). Word2Vec is a word embedding method that allows an efficient and dense representation in which similar words have a similar encoding, generated numeric representation for each word capture relations. An embedding is a dense vector of floating-point values. Generally, a higher dimensional embedding can capture fine-grained relationships between words, however, would require more computational complexity. Word2Vec has two algorithms: Continuous Bag-of-Words (CBOW) and Skip-Gram model.

In the approach of CBOW, the model predicts the current word from a window of surrounding context words, whereas the order of context words does not influence prediction. In the method of continuous skip-gram, the model uses the current word to predict the surrounding window of context words. The skip-gram architecture weighs nearby context words more heavily than more distant context words ("Word2Vec", 2020)

Doc2Vec is inspired by Word2Vec methods and creates a numeric representation of a document, regardless of its length. Since documents do not come in logical structures as words do, there was a need for a dedicated method. Doc2Vec can be viewed as an enhancement to Word2Vec, since there was a vector for Paragraph ID added to the base construct of the Word2Vec model. The feature vector is unique to each document. With word vectors being trained, a document vector is trained as well, holding a numeric representation of the document.

We apply clustering analysis on the generated document embeddings to discover specific topic clusters in the course-wide corpus. We would like to analyze whether the approach results in clustering similar movie reviews, particularly, whether there is a similarity between clustering outputs that (i) considered the introduced genre categorization and (ii) that did not.

Table 4. k-Means Clustering. Similarity Ratio. Doc2Vec vs. Doc2Vec Custom.

Iteration	Action/Adventure	Others
1	0.464	0.535
2	0.526	0.474

As with the tf-idf method, we observe a very similar alignment between the results for the Doc2Vec input matrices. The results suggest that an unsupervised machine learning method of k-Means clustering partitions data into groupings that are distinctly discriminative based on whether the evaluated text belongs to the “Action/Adventure” genre or any other one(s).

3.4 Classification Analysis. Random Forest Classification: Results

Among the algorithmic approaches, we employ Random Forests Classification for the specific text classification tasks of our research. Random Forests (RFs) are quite successful in classification and regression tasks across various applications, including practical tasks of Natural Language Processing. Random Forest is a collection of randomly constructed Decision Trees. As part of their construction, Random Forest predictors naturally lead to a dissimilarity measure among the observations. A Random Forest dissimilarity handles mixed variable types quite well, since it is invariant to monotonic transformations of the input variables, and robust to outlying and aberrant observations (“Random Forest”, Wikipedia).

Random Forest language models have the potential to generalize well to unseen data, therefore we apply a Random Forest Classifier to learn how it discriminates observations in the evaluated tf-idf and Doc2Vec matrices, specifically if the classification results demonstrate alignment with the clusters labels produced in the preceding k-Means clustering task. How and whether a Random Forest classifier distinguishes between the classes of the interest - “Action/Adventure” vs. “Others” , is of critical learning for the research objective.

We evaluated a Random Forest classifier with a cross-validation procedure. Generally, the basic approach of the cross-validation procedure is called a k-fold CV. The process takes on splitting the training set into *k* smaller sets, and for each of the *k* “folds” applying the steps as follow: 1) a model is trained using *k*−1 of the folds as training data, and 2) the resulting model is validated on the remaining part of the data. The performance measure reported by *k*-fold CV is the average of the values computed in the loop ("Cross-validation: evaluating estimator performance", scikit-learn).

Below, we present the cross-validation scores for the Random Forest Classifier across 10 iterations (cv=10). We evaluated the classifier performance for:

1) Classifier groupings vs. k-Means Clustering TF-IDF
2) Classifier groupings vs. k-Means Clustering Doc2Vec
3) Classifier groupings vs. custom k-Means Clustering TF-IDF (i.e. custom feature "Action/Adventure" vs. "Others")
4) Classifier groupings vs. custom k-Means Clustering Doc2Vec (i.e. custom feature "Action/Adventure" vs. "Others")

Table 5. Random Forest Classifier. Cross-Validation.

1	2	3	4
0.714	1.	0.857	0.143
0.714	0.857	1.	0.428
0.714	0.428	1.	0.428
0.714	0.428	0.714	0.571
0.714	0.857	0.857	0.285
0.714	1.	0.714	0.714
0.666	0.666	0.833	0.666
0.666	0.833	0.666	0.333
0.666	0.833	0.666	0.5
0.666	0.833	0.833	0.5

Average:			
0.695	**0.774**	**0.814**	**0.457**

On average, the clusters labels produced with the k-Means algorithm on the tf-idf matrix are correctly classified by the Random Forest classifier with ~70% accuracy. 77.4% accuracy is observed when the input data are represented with Doc2Vec embeddings. For the clusters with an introduced research feature of "Action/Adventure" vs. "Others", the classifier performs very well on the input data subjected to a term vectorization (tf-idf), obtaining the accuracy score of 81.4%. The classifier discriminates relatively decent on the input data transformed with Doc2Vec, demonstrating 45% accuracy in its learning.

3.5 Sentiment Analysis. DNN: Results

Among the algorithmic approaches that we evaluate in the research, we consider a deep learning model for the sentiment analysis task. We develop a neural bag-of-words model to classify the movie reviews as either those that belong to the class of "Action/Adventure" and those that belong to some other genre (i.e. "Others").

In the data preparation task, we split documents into individual words, remove punctuation, stop words, and words with a length of ≤ 1 character. We return a list of clean tokens and define a vocabulary with known words for our bag-of-words model. With vocabulary development, we aim to obtain an effective and comprehensive representation of the corpus with significant predictive power. Generally, vocabulary development is an iterative step that takes on testing different hypotheses about how to construct a useful vocabulary. We develop a vocabulary as a Counter, which is a dictionary mapping of words and their count that can allow us to easily update and query. Our vocabulary contains 794 words since we kept only words that occur at least five times in the course-wide corpus. Below are some of the words from our vocabulary that make up the top 10:

Table 6. Top 10 Predictive Vocabulary Words.

fight	*screen*	*space*	*movie*	*ambitious*	*films*	*recent*	*years*	*ideas*	*character*

With a bag-of-words method, we extract features from the corpus to acquire a required vector representation for the neural network. The number of items in the vector representing a document corresponds to the number of words in the vocabulary. Words in a document are scored and the scores are placed in the corresponding location in the representation.

Converting movie reviews to lines of tokens and encoding them with a bag-of-words model representation, we prepare the data for training of the neural network model. We use the Tokenizer class from Keras API to transform documents into encoded vectors.

We use a simple neural network architecture to predict the sentiment of encoded reviews classifying their genre, i.e. “Action/Adventure” vs. “Others”. The network construct takes on a simple feedforward network with fully connected dense layers (Dense, Keras API). The model has an input layer that equals the number of words in the vocabulary and in turn the length of the input documents. We use a single hidden layer with 50 neurons and a rectified linear activation function. The output layer is a single neuron with a sigmoid activation function for predicting “0” for “Others” and “1” for “Action/Adventure” genre reviews. The network is trained using a gradient descent Adam optimizer and the binary cross-entropy loss function, suited to binary classification problems. We gather model performance statistics during the training and evaluation phases. The model fits the training data within 10 epochs, achieving 100% in testing accuracy for three word-scoring methods and 28% testing accuracy in the frequency method.

In the research, we evaluate four different word scoring methods: binary, count, tf-idf, and frequency modes.

- In the binary mode, words are marked as a present (1) or absent (0).
- With the count mode, the occurrence count for each word is marked as an integer.
- In the tf-idf method, each word is scored based on their frequency, where words that are common across all documents are penalized.
- With the frequency mode, words are scored based on their frequency of occurrence within the document (Brownlee, 2019).

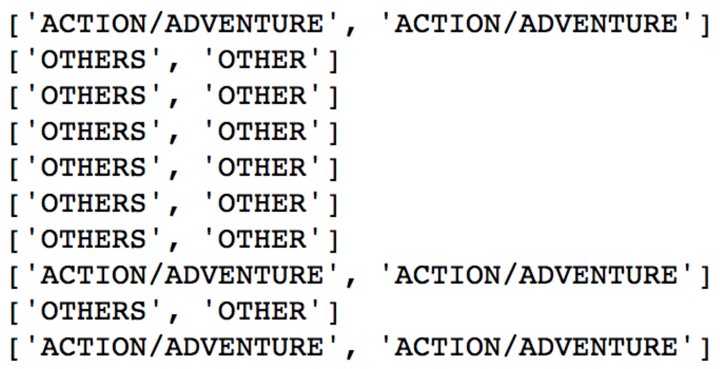
We summarize statistics for each word scoring method, assessing the accuracy means of the model skill scores across each of the 10 runs per mode.

Table 7. Word Scoring Accuracy.

	binary	count	tf-idf	frequency
Test Accuracy	1.00	1.00	1.00	0.28
Computation Time	2.20	0.49	0.58	0.43

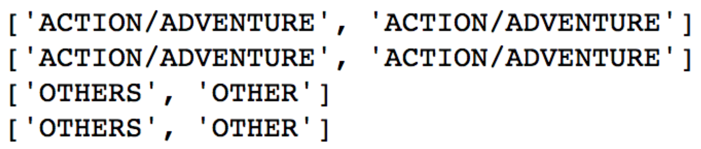
The mean score for the binary, count, and tf-idf methods appear to be equally very good. In our future networks, having all three methods performing equally well gives us flexibility in text data preparation for neural network and choosing a preferred approach for the corpus. The frequency method does not demonstrate good performance for this problem.

We perform sentiment prediction on 10 randomly selected documents from the original corpus and 4 documents from the unseen corpus to observe the model behavior and performance using the most accurate and fast performing word-scoring method (i.e. count). Our two evaluations observe absolute accuracy.



```
['ACTION/ADVENTURE', 'ACTION/ADVENTURE']
['OTHERS', 'OTHER']
['OTHERS', 'OTHER']
['OTHERS', 'OTHER']
['OTHERS', 'OTHER']
['OTHERS', 'OTHER']
['OTHERS', 'OTHER']
['ACTION/ADVENTURE', 'ACTION/ADVENTURE']
['OTHERS', 'OTHER']
['ACTION/ADVENTURE', 'ACTION/ADVENTURE']
```

Figure 1. Word-Scoring Accuracy for 10 randomly selected documents from the original corpus.



```
['ACTION/ADVENTURE', 'ACTION/ADVENTURE']
['ACTION/ADVENTURE', 'ACTION/ADVENTURE']
['OTHERS', 'OTHER']
['OTHERS', 'OTHER']
```

Figure 2. Word-Scoring Accuracy for 4 randomly selected documents from the unseen corpus.

Although the results above cannot guarantee that out models will be absolutely accurate all the time, the randomness of our selections in both tests give us an optimistic outlook on our model. There are a few improvements and extensions to consider when working on increasing the network performance: test various sizes of vocabulary, explore alternative network topologies, use regularization, apply data cleaning, evaluate training diagnostics, and consider ensemble models with different word scoring schemes (Brownlee, 2019).

4 Conclusion

Concluding the research, we would like to cover the following points:

- We develop an ontology to obtain semantic structure and knowledge of the corpus focusing on the movies of action and adventure genre that allowed us to gain a deeper understanding of the entities, objects, classes, and their relationships. The knowledge we rely on when evaluating algorithmic outputs.
- We identified high-level categorical associations (i.e. genre-based) in the evaluated corpus. That allowed us to prepare labels for our specific research question. We leveraged an introduced custom feature in the algorithmic approaches to answer our research question.
- With performance assessment of algorithmic text representation learning methods, we see how k-Means clustering, Random Forest Classification, and a simple DNN demonstrated very promising results that are closely aligned with human judgment and hypothesis with the regards to a genre of interest.
- We identified that the binary, count, and tf-idf methods can be chosen as an optimal word scoring method for the DNN model.
- Any of these algorithmic approaches can serve as a recommended method for the evaluated corpus and the specific problem. K-Means clustering and Random Forest Classification take on higher interpretability in comparison to a simple feedforward multilayer perceptron (MLP) network. However, our DNN model demonstrated top accuracy scores with almost every word scoring schema.
- For the next steps:

- The research can take on assessing multiclass classification problems considering all genres in the course-wide corpus.
- Explore an end-to-end neural network framework for text clustering.
- Evaluate the performance of the proposed algorithmic approaches for the multiclass classification and sentiment problem.
- Compare the algorithmic outputs with the human hypothesis.

5 Literature Review

In our research, we explore various algorithmic approaches in natural language processing, specifically k-Means clustering, a simple fully connected neural network, and a Random Forest classification.

Clustering is a data mining technique used to group similar items based on a similarity metric (Alsudais & Tchalian, 2019). It may be applied to different types of data like text (Alsudais & Tchalian, 2019). For text mining, clustering may be used to group text segments like words, sentences, or documents (Alsudais & Tchalian, 2019). In K-Means clustering, objects are split into different categories where each cluster contains at least one item (Alsudais & Tchalian, 2019). Each object can only be placed in one cluster (Alsudais & Tchalian, 2019). The number of clusters must be specified before initiating the algorithm (Alsudais & Tchalian, 2019). It usually takes a few trials before finding the perfect K (Alsudais & Tchalian, 2019).

Deep learning is a machine learning approach that extends the features of artificial neural networks (Wei et al., 2020). It can extract and classify features with accuracy and speed (Wei et al., 2020). Its primary goal is to classify and analyze the different patterns present in natural languages (Wei et al., 2020). It can obtain hidden features of large volumes of data automatically through the use of layers (Wei et al., 2020).

Sentiment analysis is essentially a classification problem (Alaei et al., 2017), which is defined as the placement of unseen objects into predefined categories. It can be done at a word, sentence, paragraph, or document level (Alaei et al., 2017). In binary classification, we initially assume that a review is subjective (Alaei et al., 2017). Based on the level of detail presented in the analysis, a sentiment may refer to something concrete and tangible or abstract and conceptual (Alaei et al., 2017).

The unsupervised text clustering remains a difficult and complex problem in natural language processing. Generally, such a task is approached in separate steps - text representation learning and clustering the representations. Neural methods observe an improvement with the introduction of a continuous representation learning that addresses the sparsity problem. Although, the multi-step process still deviates from the unified optimization target. In our research, we consult the paper “An end-to-end Neural Network Framework for Text Clustering” (Zhou et al., 2019) to learn the proposed pure neural framework for text clustering in an end-to-end manner. The framework jointly learns the text representation and the clustering model. Their model is evaluated on the IMDb movie reviews for sentiment classification which is highly relative to our research. Although we have taken a conventional approach to our method, we aspire to reproduce the proposed framework in future research.

6 References

1. Alaei, A. R., Becken, S., & Stantic, B. (2017). Sentiment Analysis in Tourism: Capitalizing on Big Data. Journal of Travel Research, 58(2), 175–191. https://doi.org/10.1177/0047287517747753
2. Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E., Gutierrez, J., & Kochut, K. (2017). A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques. Cornell University. Retrieved from https://arxiv.org/abs/1707.02919v2
3. Alsudais, A., & Tchalian, H. (2019). Clustering Prominent Named Entities in Topic-Specific Text Corpora.
4. An intuitive understanding of word embeddings: from count vectors to word2vec. Analytics Vidhya. Retrieved on July 16, 2020 from https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/
5. Dieng, A., Ruiz, F., & Blei, D. (2019). Topic Modeling in Embedding Spaces. Cornell University. Retrieved from https://arxiv.org/abs/1907.04907v1
6. Duan, T., Lou, Q., Srihari, S., & Xie, X. (2018). Sequential Embedding Induced Text Clustering, a Non-parametric Bayesian Approach. Cornell University. Retrieved from https://arxiv.org/abs/1811.12500v1
7. Frantzi, K., Ananiadou, S., & Mima, H. (2000). Automatic recognition of multi-word terms: The C-value/NC-value method. International Journal on Digital Libraries, 3(2), 115–130. https://doi.org/10.1007/s007999900023
8. Gensim Word2Vec Tutorial. Kaggle. Retrieved on July 16, 2020 from https://www.kaggle.com/pierremegret/gensim-word2vec-tutorial

9. Gropp, C., Herzog, A., Safro, I., Wilson, P., & Apon, A. (2019). Scalable Dynamic Topic Modeling with Clustered Latent Dirichlet Allocation (CLDA). Cornell University. Retrieved from https://arxiv.org/abs/1610.07703v3
10. Jipend, Q., Zhenyu, Q., Yun, Yuhnao, Y., & Xindong, W. (2019). Short Text Topic Modeling Techniques, Applications, and Performance: A Survey. Cornell University. Retrieved from https://arxiv.org/abs/1904.07695v1
11. K-Means clustering. Wikipedia. Retrieved from Wikipedia on August 2, 2020.
12. K-Medians clustering. Wikipedia. Retrieved from Wikipedia on August 2, 2020.
13. Li, Z., (2019). A beginner's guide to word embedding with Gensim Word2Vec Model. Retrieved from https://towardsdatascience.com/a-beginners-guide-to-word-embedding-with-gensim-word2vec-model-5970fa56cc92
14. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. Google. Retrieved from https://arxiv.org/pdf/1301.3781.pdf
15. Models.word2vec – Word2vec embeddings. Gensim. Retrieved on July 16, 2020 from https://radimrehurek.com/gensim/models/word2vec.html
16. Murdock, J. (2019). Topic Modeling the Reading and Writing Behavior of Information Foragers. Cornell University. Retrieved from https://arxiv.org/abs/1907.00488v1
17. Nijessen, R. (2017). Automatic Topic Clustering Using Doc2Vec. Retrieved from https://towardsdatascience.com/automatic-topic-clustering-using-doc2vec-e1cea88449c
18. Palachy, S. (2019). Document Embedding Techniques. Retrieved from https://towardsdatascience.com/document-embedding-techniques-fed3e7a6a25d
19. Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. Stanford. Retrieved from https://nlp.stanford.edu/projects/glove/

20. Pfeifer, D., Leidner, J. (2019). Topic Grouper: An Agglomerative Clustering Approach to Topic Modeling. Cornell University. Retrieved from https://arxiv.org/abs/1904.06483v1
21. Prabhakaran. S. Cosine Similarity – Understanding the math and how it works. Retrieved on July 16, 2020 from https://www.machinelearningplus.com/nlp/cosine-similarity/
22. Prabnu (2019). Understanding NLP Word Embeddings – Text Vectorization. Retrieved from https://towardsdatascience.com/understanding-nlp-word-embeddings-text-vectorization-1a23744f7223#:~:text=Word%20Embeddings%20or%20Word%20vectorization,into%20numbers%20are%20called%20Vectorization.
23. Random Forest. Wikipedia. Retrieved from Wikipedia on August 26, 2020.
24. Sa, L. (2019). Text Clustering with k-means. Retrieved from https://medium.com/@lucasdesa/text-clustering-with-k-means-a039d84a941b
25. Salnikov, M. (2018). Text clustering with k-means and tf-idf. Retrieved from https://medium.com/@MSalnikov/text-clustering-with-k-means-and-tf-idf-f099bcf95183
26. Scikit Learn. Cross-validation: evaluating estimator performance. Retrieved from https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation-evaluating-estimator-performance
27. Sieg, A. (2018). Text Similarities: Estimate the degree of similarity between two texts. Retrieved from https://medium.com/@adriensieg/text-similarities-da019229c894
28. Tf-idf. Wikipedia. Retrieved from Wikipedia on July 16, 2020.
29. Voronoi diagram. Wikipedia. Retrieved from Wikipedia on August 2, 2020.

30. Wang, Z., Mi, H., & Ittycheriah, A. (2016). Semi-supervised Clustering for Short Text via Deep Representation Learning. Cornell University. Retrieved from https://arxiv.org/abs/1602.06797v2

31. Wei, W., Wu, J., & Zhu, C. (2020). Special issue on deep learning for natural language processing. Computing, 102(3), 601–603. https://doi.org/10.1007/s00607-019-00788-3

32. Word2vec. Wikipedia. Retrieved from Wikipedia on July 16, 2020.

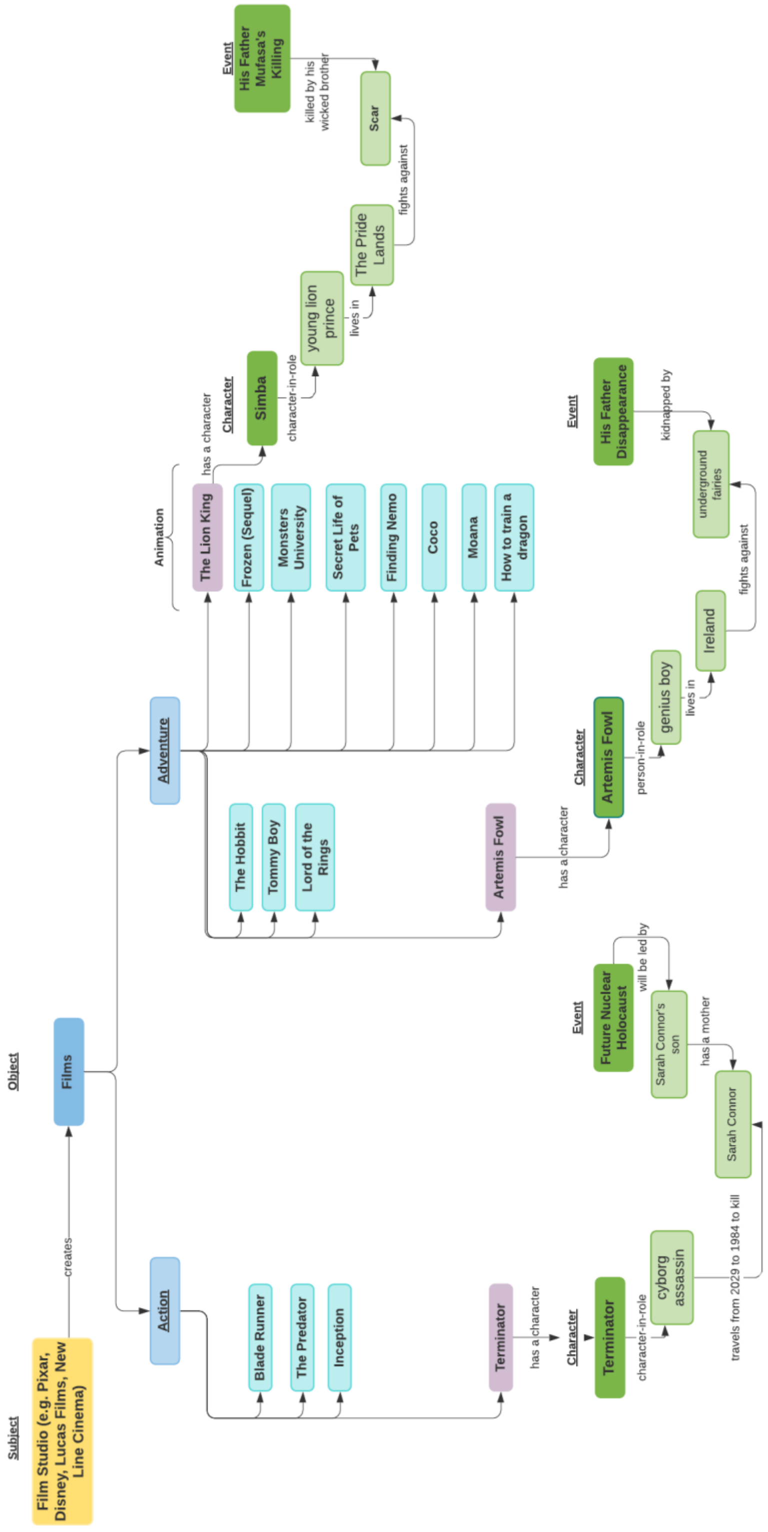
33. Xu, B., Guo, X., Ye, Y., & Cheng, J. (2012). An Improved Random Forest Classifier for Text Categorization. Retrieved from https://pdfs.semanticscholar.org/99ef/762bebb86811f83d626d78feb66f31262ccd.pdf

34. Xu, P., Jelinek, F. (). Random Forests in Language Modeling. Center for Language and Speech Processing the Johns Hopkins University. Retrieved from https://www.aclweb.org/anthology/W04-3242.pdf

35. Zhou, J., Cheng, X., & Zhang, J. (2019). An end-to-end Neural Framework for Text Clustering. Retrieved from https://arxiv.org/pdf/1903.09424v1.pdf

7 Appendix

7.1 Ontology



7.2 Student Contributions to the Research

Yabo Gao	Julianna Antonchuk
• Code development: k-Means Clustering, Random Forest Classification, Sentiment Analysis • Results Interpretation • Literature Review • Paper Peer Review	• Ontology development for the Action and Adventure genre of the evaluated course-wide corpus • Research Paper development: composition, structure, grammar, formatting, readability, references, etc. • Results Interpretation • Literature Review