UNIVERSITY OF CALIFORNIA, SAN DIEGO

Bag-of-Concepts as a Movie Genome and Representation

A Thesis submitted in partial satisfaction of the requirements

for the degree Master of Science

in

Computer Science

by

Colin Zhou

Committee in charge:

Professor Shlomo Dubnov, Chair
Professor Sanjoy Dasgupta
Professor Lawrence Saul

2016

The Thesis of Colin Zhou is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

____

____

____

Chair

University of California, San Diego

2016

DEDICATION

For Mom and Dad

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

ABSTRACT OF THE THESIS

Bag-of-Concepts as a Movie Genome and Representation

by

Colin Zhou

Master of Science in Computer Science

University of California, San Diego, 2016

Professor Shlomo Dubnov, Chair

As online retailers, media providers, and others have learned, the ability to provide quality recommendations for users is a strong profit multiplier: it drives sales, maintains retention, and provides value for users. Video streaming, retail, and video databases particularly have had extensive effort and research devoted to finding effective ways to make representations. One of these is MovieLens.org, a video recommender website that uses a representation called the tag genome to understand movies and make tuned recommendations. The tag genome representation is built from user input on the website, where users can apply tags to movies and rate them in order to provide the information needed to make recommendations.

In this work, we draw from research in information retrieval to implement a bag-of-concepts representation for movies and make tuned recommendations in the same manner as MovieLens. Our implementation is fully unsupervised and does not require the user data needed in the implementation of MovieLens while still having similar properties to the tag genome that enable interesting tuned recommendations.

Introduction

Online movie/video information databases and video streaming services often come with a recommendation feature that finds new videos that users may enjoy. These recommendations can keep users engaged and improve their experience with the site or service, which makes these recommendation systems an important feature for designers to focus their efforts on. The Netflix Prize challenge [20] shows the interest in and importance placed on recommender systems by researchers and companies. The competition's goal was to find a recommender system that could beat the accuracy of Netflix's own system by ten percent. Netflix offered a prize of one million dollars and gathered significant media attention, and more than forty thousand teams submitted their own recommender systems.

Many recommender systems for movies that are discussed in literature require both detailed information about the films themselves as well as a body of user data and feedback in order to make their recommendations. This work is an attempt at implementing and evaluating a recommender system that has much lower data requirements: we use only text descriptions of each movie's plot and no user data in order to build representations for our movies and make recommendations. The data set used in this paper comes from IMDB.com, accessed through its public FTP interface [3]. Plot descriptions are contributed by its userbase, and contain a large amount of variation.

Users differ in their writing style, tone, descriptiveness, quality, and more when writing plot summaries, which makes the dataset a challenging one to work with.

In Chapter 1 we review relevant literature from the areas of document representation and recommender systems. Models of representation discussed are the vector space model and neural net derived distributed representations. The two approaches of collaborative and content-based recommenders are explained, and relevant previous work is reviewed. In Chapter 2 we describe the methods of manipulating bag-of-concept vectors with meaningful semantics in their composition as an analogue of tag genome navigation pioneered by the GroupLens research lab [27]. Chapter 3 evaluates the performance of bag-of-concepts in comparison to other representation methods in the task of genre classification. The paper concludes with a discussion of possible future work in chapter 4.

Chapter 1

Background

Document representation has always been a important area of work for fields such as information retrieval and data mining. Once computer systems were put to use in storing and retrieving large bodies of information, researchers began looking for ways to automatically search for documents in a large collection. One influential idea that emerged in this early era, published by H.P. Luhn in 1957 [4], was that the encoding of a document could be made automatic, and that it could capture the "concepts" in a document without needing to understand it in the same way that a human cataloger would. Luhn proposed that the conceptual makeup of documents could be indicated by the words in the document, and that documents could be retrieved based on the similarity of their word distributions.

1.1 Vector Space Models

One key development of the 1960's was the SMART information retrieval system, developed by a team led by Gerard Salton[5,6]. The SMART system introduced the vector space model [7, 8] for representing documents, which has become a very widespread model for representing documents and objects in general. In the vector space model, documents are represented as vectors with each dimension corresponding to a different term. In most cases a term would be a single word, but in different applications a term can be a phrase or other unit. If a term appears in the document, then its corresponding dimension in the representation has a non-zero value [41]. When these vectors are considered as points in vector space, the semantic similarity of documents can be compared by computing the angle between two vectors (aka the cosine distance).

1.1.1 Bag-of-Words

Two examples of vector space models are bag-of-words and tf-idf (short for term-frequency, inverse-document-frequency). In bag-of-words, the value of a vector dimension is simply the count of it's term's appearance in the document. The concept of a bag-of-words representation has actually existed since before the SMART team proposed the vector space model (referenced in 1954 by Z.S.Harris [9]) and, while simplistic, is still in widespread use.

1.1.2 TF-IDF

Tf-idf, as its name describes, uses both the term frequency (word count) and the inverse document frequency (percentage of documents in the corpus a term appears in) to determine the value each term has in the vector space model representation [10]. A common implementation in practice is to multiply the term frequency by the log of inverse document frequency. For example, the tf-idf value of term *t* in document *d* may be calculated as:

$$tfidf(t,d) = \frac{d_t}{\sum_{i=1}^{T} d_i} \cdot \log \frac{D}{\sum_{j=1}^{D} 1\{j_t \neq 0\}}$$

where d_i is the count of term *i* in document *d*, *T* is the total number of terms, and *D* is the total number of documents.

1.1.3 Latent Semantic Indexing

Published in 1988, latent semantic indexing [39](also called latent semantic analysis) is a technique for reducing the size of the representation needed for a dataset without greatly impacting the quality of said representation. LSI begins with a matrix of weighted term occurrences for a dataset. Rows represent different terms and columns represent different documents, with values in each cell for weighted counts. This matrix can be viewed as a vector space model representation of all the documents. LSI then uses singular value decomposition to find a truncated decomposition of the original matrix. While the results of LSI are much more compact, the dimensions are no longer independent and tied to a single term anymore.

1.2 Neural Net Models

Work in the field of neural nets have also led to methods of representation based on the values inside the net itself. In 1986, Geoffrey Hinton published a paper [11] that showed how neural nets can be made to learn patterns in its nodes that are distributed representations of concepts. By distributed representations, we mean that the meaning of the represented concept is not localized to a particular dimension as it is in vector space models, but distributed across the entire vector. This allows any dimension to be non-zero in a distributed representation of a concept. These distributed representations, or embeddings, come from the learned weights in a neural net that has been trained to recognize different inputs.

Scientists in the field of cognitive science and computer science would apply neural nets to model language [12], training them to learn word embeddings. In these word embeddings, words that occurred in similar contexts are considered similar, and have vectors that are close together in the representation vector space. This property is a great advantage over vector space models, where each term is considered completely independent and there is no notion of similarity among terms.

Another advantage of these embeddings is that their size is not bound to the number of terms in the dataset, a property of vector space models that led to quickly diminishing returns as the size of the dataset grew. Needing representations the size of a dataset's vocabulary becomes rather cumbersome, and the quality of any measure of similarity suffers due to the curse of dimensionality with large, sparse vectors. Instead, the size of neural net representation is tied to the size of the neural net itself (the number of nodes in the network). Neural nets only need to be of sufficient size to achieve the desired level of accuracy in distinguishing inputs, which is much smaller than the number of terms in a dataset. Neural net embeddings are not without their own drawbacks, as training embeddings is a relatively slow and computationally intensive process, especially for the time they were proposed in.

1.2.1 Neural Net Architectures

In the early 2000s, increasingly powerful computer hardware and advances in neural nets allowed them to be deployed on larger scales, which sparked more interest in using them to model language again. During this period many different methodologies for training neural net embeddings were proposed and evaluated. In 2003 Bengio et al. [13] proposed a feed-forward neural net architecture that learned word embeddings and a statistical model of the distribution of words. Feed-forward architectures have connections that only move forward from the input layer towards the output layer, and do not form cycles. The feed-forward net is trained to estimate the conditional probability of the next word given the a window of some number of previous words and the distributed representation vectors in the same process.

In 2010, a recurrent neural net architecture for language modeling was proposed by Mikolov et al. [14]. Unlike feed-forward architectures, recurrent networks have directed cycles that are able to serve as a form of memory across time. For language modeling, this is the sequence of words in the documents. This memory allows recurrent nets to calculate word probabilities taking into account all previous words instead of a limited window [34]. Recurrent nets were able to match the performance of feed-forward nets while using less training data, and achieved higher accuracy than other published methods at the time [35].

Currently one of the most popular family of models is known as word2vec, introduced in 2013 by Mikolov et al. [15, 16]. The general ideas of the word2vec model have also been extended to create embeddings for entire documents, called paragraph2vec or doc2vec [17]. The neural net architecture used for these models is exceedingly simple, a single fully-connected hidden layer between the input and outputs. Yet, the representations learned by word2vec were of visibly higher quality than existing methods. How was this the case?

The simplicity of word2vec and doc2vec's neural net allows it to be trained very fast compared to other language models, and its speed means that it can feasibly be run on much larger datasets than previous models. Word2vec demonstrated that, for a given period of time, it is better to run a simpler model on a larger dataset than to use a more sophisticated model on a smaller dataset. With the rise of big data, large datasets were easier to find and so word2vec and doc2vec became very popular models to use for vector representations.

A very interesting property of neural net word embeddings is that the learned vector representations actually encode many meaningful semantic relationships that allows for reasoning and analogy in the form of vector operations [16, 18]. For example, the vector combinations of “King – Man + Woman” results in a vector that is very close to “Queen.” These examples demonstrate that directions in the representation's vector space can actually be meaningful and represent relationships or concepts.

One disadvantage of word2vec and doc2vec representations is that, although it can capture a lot of semantic information, the vectors themselves have no real human interpretation. Each value comes from the connection weighting in the trained neural net, which is rather meaningless to a human. By contrast, values in a bag-of-words vector are easily understood as the count of occurrences in a document. Even for other weighting schemes among vector space models, values can be understood as some measure of the significance of a term for the given document.

1.3 Bag-of-Concepts Model

The bag-of-concepts model (Kim et al. 2015 [1, 2]) is an attempt to combine the advantages of learned similarity from neural net embeddings and human interpretability from vector space models. The model learns word vectors from documents in the fashion of word2vec, and then clusters vectors into "concepts" using cosine distance and spherical k-means clustering. Words that are similar are clustered into the same concept due to the closeness of their word2vec vectors. Using these clusters, a vector space model is built from the document where each term is a concept cluster, with all the words in a concept cluster being counted as occurrences of the term in a document. As a model that ultimately results in a reduced-size vector space model, the bag-of-concepts model has parallels to LSI in representation size reduction, but has an advantage over the latter in that it is still an actual vector space model and thus more easily interpretable to humans than a LSI-reduced representation.

1.4 Recommender Systems

1.4.1 Collaborative Recommenders

Recommender systems can be classified as either using a collaborative or content-based filtering approach, or a hybrid of the two approaches. Collaborative filtering recommender systems rely on user data or contributions in order to make recommendations. Users can leave ratings, likes, dislikes, or other feedback that ties together items with similar responses. The strengths of collaborative systems are their universal applicability and ability to find links between otherwise dissimilar items [21]. Because these systems make recommendations based on user feedback, they have no need to analyze and understand the actual items themselves, and can be put to use in recommending any category of items possible. User data is also filled with item relationships that wouldn't be found by considering similar items: Flashlights and batteries are nothing alike, but it wouldn't be hard to find the two being bought together by customers. Collaborative recommender systems have been successfully implemented in many popular commercial platforms, such as Ebay (user ratings), Amazon ("Customers who bought this item also bought" widget), and Facebook (likes) [22, 23].

One of the drawbacks of collaborative recommenders is known as the "cold-start" problem [24], when there is not enough user feedback to link entries in a dataset together. This can come from both entirely new systems with no feedback to base any recommendations on, or from established systems that have new, undiscovered, or rare entries without the feedback to link them to other entries. To address this problem, commercial recommender systems use a variety of hybrid approaches that use content-based recommendation techniques in addition to collaborative recommendation.

1.4.2 Content-based Recommenders

Content-based approaches avoid the “cold-start” problem of collaborative approaches by basing their decisions entirely on the intrinsic properties of each entry. For example, a movie's genre, runtime, rating, and starring actors can all be properties that a content-based recommender would consider to make movie recommendations. This approach draws from many aspects of research done in the information retrieval community, where document representation techniques can more abstractly capture the features of an item to make comparisons against potential recommendations [25].

When user feedback is used to personalize recommendations, a content-based recommender can actually build a profile for an individual user that captures personal taste for customized recommendations [29]. The most common method for doing so involves learning a set of weights for each property of a content-based representation from what a user has liked before. Properties that users seem to care about learn higher weights, and those that don't seem to affect their decisions learn lower weights. This allows customized recommendations that should align with a user's tastes because they are similar in the specific qualities that that user cares about.

1.4.3 Recommendation Through “Genes”

Inspiration for further research into identifying and indexing all of the different features and aspects of the items in a recommendation dataset came from an unlikely source: the Human Genome Project. Just as all of the genes in human DNA were to be mapped and identified in the Human Genome Project, so too were researchers inspired to find and catalog the building blocks describing their datasets. The Music Genome Project developed and copyrighted by Pandora Radio [30], the Movie Genome (now called Entertainment Genome) developed for the Jinni movie search and recommendation engine [31], and the tag genome developed by the GroupLens research lab for their MovieLens recommender [26, 27, 28, 32] all seek to find the "genes" of their respective domains (though biologists might argue that the analogy is closer to phylogeny and thus should be renamed "Phenome Projects").

These "genome projects" improved the quality of data representations for entries in their domain by using the domain-specific genes, compared to the more generic representations initially adapted from information retrieval and filtering research. The "genome" style of representation had a great advantage in human interpretability. Recommenders could point to specific genes to explain *why* items were chosen: Bob might seem to be a fan of space travel, so *Star Trek* was recommended for that reason.

The genome representations also supported a novel method of *navigating* a dataset, by moving along the dimensions represented by genes [27]. The tag genome is created by running a supervised learning process on the user-applied tags, ratings, and reviews against a ground truth of user survey results, where participants were asked to rate the relevance of a set of tags to movies. After training, the tag genome for a movie is the vector of relevance scores for all tags in the genome as they are scored for that movie

[26]. The GroupLens research group has demonstrated this type of navigation in depth with their Movie Tuner interface to the tag genome and the MovieLens website [33]. The Movie Tuner interface asked users to critique a query movie and made suggestions based on their answers. If a user is asked about the movie “Iron Man” and requests a movie that is “more gritty,” he might be suggested “Batman Begins.” The tag genome representation capture how strongly a tag relates to a movie, and can explore movies that have a stronger or weaker relation according to the user's wishes. This navigation parallels many of the ways people think of movies and music, with analogies of the form “Movie A is a lot like Movie B, but with more of Gene X.” When a genome has been compiled that enumerates all of the “gene” elements as dimensions of the dataset, human users are enabled to search and navigate through dimensions because of the semantic meaning each dimension has.

Chapter 2

Concept Navigation and Tuning

Our motivation for this research was to explore movie representation and recommendation in a fashion similar to the tag genome representation and movie tuner recommender developed by GroupLens, but in an unsupervised manner and without the user-supplied data requirement. For this purpose, we test the bag-of-concepts representation model for suitability.

2.1 Methodology

At the time of access, the IMDB database had plot descriptions for 262,903 distinct movies, tv series, or shorts, with some having multiple different plot descriptions submitted by different users for a total of 449,562 separate plot descriptions. For episodes of a TV series, all plot descriptions of individual episodes are united under one label for the series as a whole. The text descriptions were preprocessed by using the NLTK pre-trained Punkt tokenizer for English [19]. For our implementation of the bag-of-concepts vector representation, the word2vec word vectors were trained with a size of 100, then clustered into 100 concept groups.

2.2 Concepts

The word vector clusters, or “concept” clusters, are somewhat mixed but demonstrate some promise. While some clusters do not seem to have a common theme to them, there are just as many clusters that do. The fact that at least half of the clusters we examined had a human-intelligible theme demonstrates a good deal of success. Table 1 shows some example clusters and a few words in each, as well as a human “best guess” as to the topic of each cluster.

Table 1: Some words from selected word clusters and a human “best guess” for the cluster topic

Some words from Cluster	Topic “Best Guess'
'compulsory', 'gcse', 'filibuster', 'poltroni', 'ballot', 'negotiating', 'greenlight', 'doctors', 'institution', 'secretarial', 'dossier', 'diplomatic', 'project', 'shakeup', 'legalizing', 'enforcement'	Government/Management
'orchestrating', 'thorton', 'criminal', 'tienbao', 'gunning', 'peskin', 'lunatic', 'seeking', 'goon', 'killer', 'arsonist', 'wellorganized', 'diabolik', 'framed', 'extortionist'	Evil/Villainous
'wormhole', 'breathable', 'girdle', 'xrays', 'neural', 'nanobots', 'outerspace', 'exorcism', 'computer', 'dinosaurlike', 'pathogen', 'dispenses', 'hologram', 'extracting', 'ingenious'	Scientific/High-Tech
'ligament', 'decease', 'amyotrophic', 'underwent', 'ballooned', 'ulcers', 'hominis', 'haeme', 'chromosomes', 'hemorrhage', 'bruising', 'pulmonary', 'mardon', 'mammogram', 'anguishes', 'hepatitis', 'hemorrhages'	Medical/Diseases
'bharat', 'asit', 'jaggannath', 'singhal', 'dayashankar', 'matondkar', 'anastasias', 'yodhraj', 'deepa', 'yadav', 'singhanias', 'mullick', 'bhavani', 'karim', 'sudhirbhai'	Indian names
'ingratiates', 'alerts', 'importunes', 'whereat', 'sics', 'cashes', 'compares', 'obliges', 'administers', 'vanquishes', 'overwhelms', 'derails', 'grasps', 'alleviates'	Verbs, third person present tense
'internalizing', 'democratization', 'pericles', 'morbidity', 'updating', 'outline', 'reproduction', 'reverberations', 'delineates', 'nicolais', 'microbial', 'hooping', 'metaphysics', 'roiling', 'polarization', 'hatfields'	(No sensible topic)

One detraction from the usefulness of the concept clusters is that currently there is still the need for a human to look at the words in a cluster to make an educated guess as to the theme of the cluster. Automatic labeling of topic models is an area of current research that could be applied to the concept clusters for labels in future work [40].

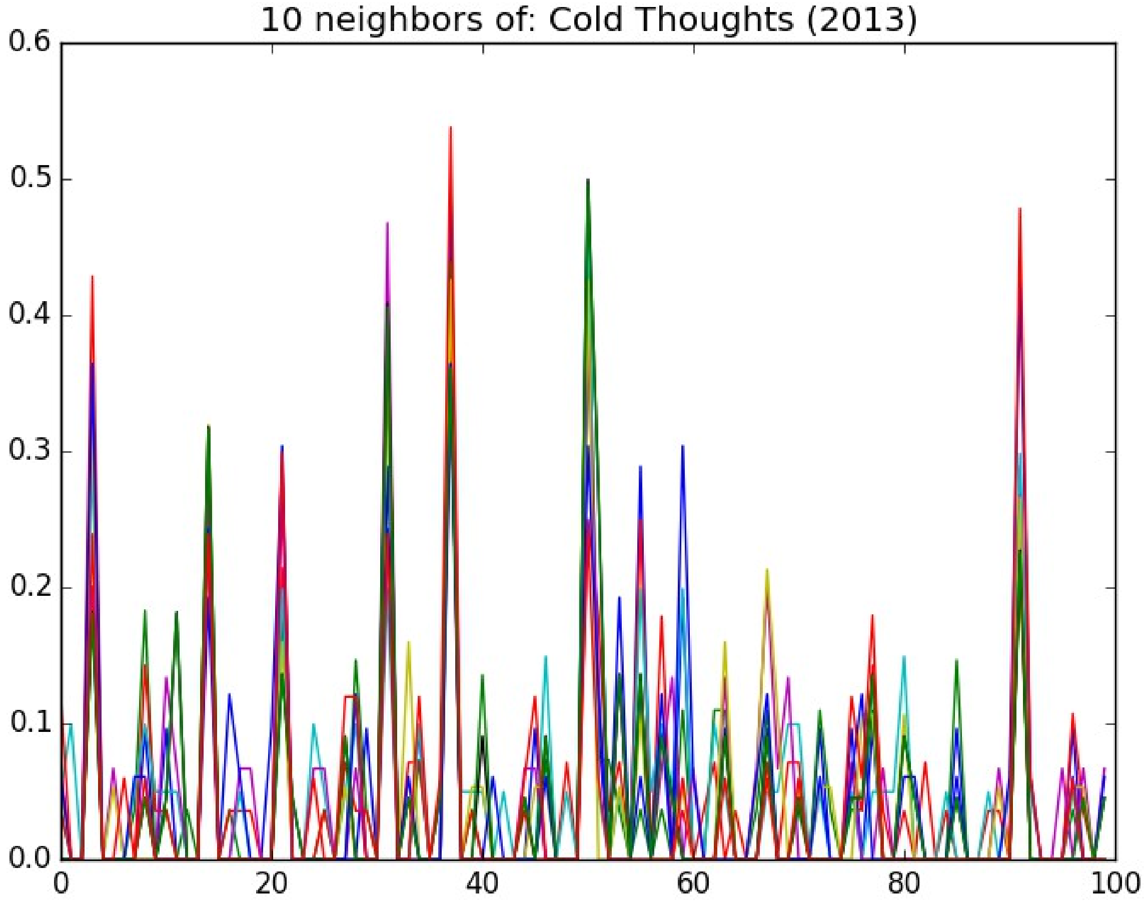

Figure 1: Ten BOC Vectors for the Nearest Neighbors of *Cold Thoughts (2013)*

Figure 1 shows ten bag-of-concepts vectors each represented as a line over its 100 concepts. The ten films represented here are the ten nearest neighbors of a random film from the dataset. From the figure there are noticeable peaks which represent a significant part of each film's concept makeup. These significant concept peaks change from film to film, and we hypothesize that the makeup of these significant peaks can be a strong indicator for the kind of plot in a film.

2.3 Recommendation Tuning

The bag-of-concepts representation can be directly manipulated through changing its cluster values. Through these changes, we can navigate the space of our representation in a understandable way. One way to do this is to raise or lower a specific cluster value. As an example, we use the bag-of-concepts vector for a description of the 1999 film *The Matrix*, and increase the count of the "Government/Management" cluster (given as example in Table 1). After normalizing, we look at the top five nearest neighbors of this modified vector in order of increasing distance:

The Matrix (1999)
A Very British Coup (1988)
The Matrix (1999)
A Very British Coup (1988)
Yeon-ga-si (2012)

It's not surprising that two different descriptions of *The Matrix* show up as nearest neighbors, as the modified vector shares most of its cluster value distribution with the original description of *The Matrix*. *A Very British Coup* describes a fictional political maverick elected to office who begins to butt heads with sinister power brokers in the upper echelons of the UK. *Yeon-ga-si* is a Korean thriller about a series of infections breaking out, and the investigations of the deaths leading to uncover a conspiracy between the government and a pharmaceutical company about its origins and cure.

A plausible explanation for the link between each of these neighbors would be that they all feature some sort of dark, hidden conspiracy (whether it be orchestrated by AI machines or villainous humans) to control the rest of the population. While it seemed odd that neither of the other nearest neighbors have anything approaching the slick gunfights and action scenes of *The Matrix*, this may be explained by the particular plot description of *The Matrix* that we chose to modify focusing more on the reveal of the machine conspiracy rather than putting an emphasis on describing the action fights. The nearest neighbors we found also have more elements of the "Management/Government" concept compared to *The Matrix*, with human governments replacing the machines as hidden conspirators.

In another example, we use the *James Bond* spy film *GoldenEye* as a base and look for movies with more "medicine/disease" content, as represented by the cluster shown in Table 1. In this example, the movies found similar to "*GoldenEye* + *Medicine/Disease*" are, in order of increasing distance:

GoldenEye (1995)
- James Bond must stop a satellite weapon system from falling into the wrong hands

The Art of War III: Retribution (2009)
- A UN agent must stop North Korea from using a nuke on a UN peace

summit

Outbreak (1995)

- U.S. Army medical researchers race to stop a deadly virus outbreak and a general with ulterior motives

Die Another Day (2002)

- Another *James Bond* film that features the villain undergoing plastic surgery

Sheng hua te jing: Sang shi ren wu (2000)

- A Hong Kong Cops-vs-Zombies comedy/horror film

In this example, the order of the list of results leads to an interesting possible insight. *GoldenEye* and *The Art of War III* are quite similar in their plots, without any trace of the medicine or disease concept present in either. *Sheng hua te jing* seems to be a bit too far from *GoldenEye* in its plot and style, but the disease concept is well-represented. *Outbreak*, in the middle of the list of results, can be seen as the best "quality" recommendation that has elements of both *GoldenEye*'s thriller plot style and the medical/disease concept. While the quality of our concept manipulation analogies is ambiguous to judge, it seems that there is a "sweet spot" for these recommendations not too far or too close from our query vector that yields the best results.

2.4 Discussion

The previous examples demonstrate that the bag-of-concepts document representation method can be used with concept tuning for exploring representation spaces. Our results show similarity to tuned recommendations in the tag genome, with concepts working as analogues to tags. The current quality of our bag-of-concepts representation is still below that of the tag genome, as judged by the proportion of clusters without a clear concept theme, occasional unexpected words in concept clusters (such as 'exorcism' in the otherwise science/high-tech-themed cluster), and the “quality” of some tuned recommendation examples not shown here. While there is not a clear standard for how to judge the quality of recommendations, we based our judgement on our impressions and believe that a future user study of a larger group could serve as a good foundation on which to base relative judgements of quality. We speculate that a significant step to improving the perceived quality of recommendations is to find the correct “sweet spot” for our calculation of tuned recommendations, particularly in determining how far to move in a concept direction.

Among our concept clusters, the themes that we found were quite varied and contrasted with user-applied tags (as ones in the tag genome) in interesting ways. While some clusters can be considered similar to human tags (such as a scientific/high-tech cluster), others have topics that wouldn't generally be considered tags, such as parts of speech or names. Human tags are applied by users with the framework of identifying and comparing movies in mind, which means that the tags are considered useful for that purpose in the first place.

The concept clusters from our bag-of-concepts implementation are generated without such purpose in mind, and some do not seem like they will be helpful in comparing movies or making recommendations. It may be that these clusters are actually useful for capturing the properties of films and do help in making recommendations in a non-obvious way, but one of our main goals was to maintain interpretability in our implementation: if a concept cluster does not make sense to people then it fails to reach that goal regardless. While bag-of-concepts was judged as not as consistent as MovieLens's tag genome in being human-interpretable so far, this is not unexpected for a proposed unsupervised process versus a supervised one.

Chapter 3

Genre Classification

We would like to compare the quality of the bag-of-concepts representation with other document representations in an objective analysis. As common examples of vector-space and neural net-based models, we will use tf-idf and doc2vec as our comparison representations. We compare these representations in the task of genre classification. In our dataset, movies can be labeled with multiple genres, such as a sci-fi, action, and thriller movie, or a romance and drama movie. This is a multi-label classification problem [36, 37], which we approach in a baseline binary relevance method. Separate classifiers are trained for each genre in a one-vs-all manner. The classifier used in all cases was a SVM with a linear kernel implementation from scikit-learn [38], which was selected over other alternatives due to limited computational resources.

3.1 Methodology

Our 449,562 separate plot descriptions are split into 90% training and 10% test sets, then transformed into the three different representations we will evaluate. For our implementation of the bag-of-concepts vector representation, the word2vec word vectors were trained with a size of 100, then clustered into 100 concept groups. The doc2vec vector representations were trained with a vector size of 300. tf-idf weighting was applied to the bag-of-words vectors of each entry.

Among all of the genre labels given in our dataset, we focus on the top ten by popularity. These are: 'Drama', 'Short', 'Family', 'Crime', 'Romance', 'Adventure', 'Action', 'Comedy', 'Documentary', and 'Thriller'. Separate classifiers are trained for each genre, for a total of ten one-vs-all classifiers for each representation. Predicted labels are evaluated against the genre labels from our dataset, using F-score as a performance metric.

3.2 Results

Table 2 lists the average F-scores over all genres for each method. Table 3 in the appendix lists a more detailed breakdown by genre.

Table 2: Precision, recall, and F-score averages over all genres

Representation	**Precision**	**Recall**	**F-score**
tf-idf	0.5142	0.7637	0.6079
doc2vec	0.3785	0.7266	0.4839
Bag-of-Concepts	0.3701	0.7283	0.4775

The bag-of-concepts and doc2vec representations both yield comparable F-scores, with the tf-idf representation coming out ahead of both. While the bag-of-concepts representation has a slightly lower F-score than doc2vec, we believe the interpretability of bag-of-concept vectors and their manipulations can be an advantage over doc2vec to anyone interested in exploring and understanding the dataset. Two variations to the bag-of-concepts representation were also tested, by applying PCA and tf-idf to the bag-of-concepts word counts instead of normalizing. Results for the two variations were very slightly worse than the original normalized bag-of-concept vectors.

3.3 Discussion

It is interesting to note the higher performance of tf-idf compared to vector-based approaches, which differs from previous work that found word vectors to beat state-of-the-art in sentiment analysis. We speculate that different results are due to the difficulty of the problem and specific nature of terms and genres.

It may be that the grouping of words into clusters is actually detrimental to genre classification if very strong genre-signifying words become mixed in with more genre-generic words within clusters. For example, the words 'computer' and 'electric' would not strongly indicate that the plot describes a science fiction movie, but the words 'exoskeleton' and 'nanobots' would make it much more likely. However, our word clustering actually does put all of the previous four words in the same cluster, treating them the same for the purpose of representation.

In addition to being a multi-label problem with ten categories compared to sentiment analysis's binary positive or negative, the difference between genres may be quite subtle, sometimes even to human reviewers. Descriptions of movies in separate genres may share a lot of common vocabulary, with just a few significant words that could be used for distinction. This mostly impacts classification precision scores, which are much lower in our results than recall scores.

Chapter 4

Future Work

The current quality of our bag-of-concepts representation is still below that of the tag genome implemented by GroupLens. This is true for both the quality of concepts as well as tuned recommendations. While this is to be expected, we believe there are steps we can take towards the standard set by the tag genome while still relying on only the movie description dataset. The first step would be to consider more defined ways of measuring and talking about the quality of recommendations. Judgements of quality are made by the researcher's impressions currently, and a more objective measure or wider pool of judges would be better. While recommendations that use the same dataset are easy to compare in quality, we are using a different dataset, both in the items in the set and the information about each item. There is no obvious measure of quality or point of comparison, which leaves user test studies as the remaining option. A user experiment could be organized as a survey asking participants to rate the relevance of tags to movies or the quality of a tuned recommendation.

Secondly, as mentioned before, we would like to use a method of automatically labelling each concept cluster with a topic. Currently it is the researcher's best guess that labels each cluster. While this is likely more accurate than automatic labeling will be, it is slow and not in the spirit of a fully unsupervised process that we are aiming for.

The quality of our tuned recommendations showed interesting results that may lead to insight on what is needed for a high-quality recommendation. We noticed some examples of a “sweet spot” for recommendations that were not too similar or dissimilar. Whether this sweet spot is consistent and if it can be learned in a supervised or unsupervised fashion is still unknown and exists as potential future work.

In addition to our current method of creating tuned recommendations, we would also like to explore other possible types of concept vector manipulations for recommendations. As an example, does averaging two or more movie's concept vectors together lead to recommendations that feel like a mix of the two movies? Or, can we make a concept vector from a word or sentence input from a user and use the result as we would a movie base or concept modifier in recommendations? Another interesting possibility is to see if our concept vectors can match the compositionality and vector operations that word2vec and doc2vec are known for.

Appendix

Table 3: Precision, Recall, and F-1 Scores broken down by genre

Genre		TF-IDF	Doc2Vec	BOC
Drama	**Precision**	0.7476	0.6599	0.6594
	Recall	0.7693	0.7154	0.7339
	F-1 Score	0.7583	0.6865	0.6946
Short	**Precision**	0.5603	0.4314	0.4354
	Recall	0.7733	0.6948	0.7338
	F-1 Score	0.6498	0.5323	0.5465
Family	**Precision**	0.3870	0.2387	0.2133
	Recall	0.7165	0.6774	0.6822
	F-1 Score	0.5026	0.3530	0.3249
Crime	**Precision**	0.4899	0.3498	0.3770
	Recall	0.7815	0.7634	0.7345
	F-1 Score	0.6023	0.4797	0.4983
Romance	**Precision**	0.3605	0.2255	0.2195
	Recall	0.6941	0.6664	0.6892
	F-1 Score	0.4745	0.3370	0.3330
Adventure	**Precision**	0.3786	0.2450	0.2356
	Recall	0.7245	0.7240	0.6906
	F-1 Score	0.4973	0.3661	0.3513
Action	**Precision**	0.4779	0.3153	0.3112
	Recall	0.7761	0.7496	0.7371
	F-1 Score	0.5915	0.4439	0.4376
Comedy	**Precision**	0.6662	0.5251	0.4829
	Recall	0.7925	0.7172	0.7022
	F-1 Score	0.7239	0.6063	0.5723

Table 3: Precision, Recall, and F-1 Scores broken down by genre, continued

Genre		**TF-IDF**	**Doc2Vec**	**BOC**
Documentary	**Precision**	0.6808	0.5286	0.5191
	Recall	0.8768	0.8248	0.8665
	F-1 Score	0.7665	0.6443	0.6493
Thriller	**Precision**	0.3933	0.2659	0.2475
	Recall	0.7325	0.7335	0.7130
	F-1 Score	0.5118	0.3904	0.3675

References

[1] Kim, Han Kyul, Hyunjoong Kim, and Sungzoon Cho. "Bag-of-Concepts: Comprehending Document Representation through Clustering Words in Distributed Representation." (2015).

[2] Kim, Han Kyul, Hyunjoong Kim, and Sungzoon Cho. "Distributed Representation of Documents with Explicit Explanatory Features."

[3] Alternative Interfaces. URL http://www.imdb.com/interfaces

[4] Luhn, Hans Peter. "A statistical approach to mechanized encoding and searching of literary information." *IBM Journal of research and development* 1.4 (1957): 309-317.

[5] Salton, Gerard. "The SMART retrieval system—experiments in automatic document processing." (1971).

[6] Buckley, Chris. *Implementation of the SMART information retrieval system*. Cornell University, 1985.

[7] Salton, Gerard, Anita Wong, and Chung-Shu Yang. "A vector space model for automatic indexing." *Communications of the ACM* 18.11 (1975): 613-620.

[8] Turney, Peter D., and Patrick Pantel. "From frequency to meaning: Vector space models of semantics." *Journal of artificial intelligence research* 37.1 (2010): 141-188.

[9] Harris, Zellig S. "Distributional structure." *Word* 10.2-3 (1954): 146-162.

[10] Sparck Jones, Karen. "A statistical interpretation of term specificity and its application in retrieval." *Journal of documentation* 28.1 (1972): 11-21.

[11] Hinton, Geoffrey E. "Learning distributed representations of concepts." *Proceedings of the eighth annual conference of the cognitive science society*. Vol. 1. 1986.

[12] Miikkulainen, Risto, and Michael G. Dyer. "Natural language processing with

modular PDP networks and distributed lexicon." *Cognitive Science* 15.3 (1991): 343-399.

[13] Bengio, Yoshua, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. "Neural probabilistic language models." *Innovations in Machine Learning*. Springer Berlin Heidelberg, 2006. 137-186.

[14] Mikolov, Tomas, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. "Recurrent neural network based language model." *INTERSPEECH*. Vol. 2. 2010.

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

[16] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013.

[17] Le, Quoc V., and Tomas Mikolov. "Distributed representations of sentences and documents." *arXiv preprint arXiv:1405.4053* (2014).

[18] Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. "Linguistic Regularities in Continuous Space Word Representations." *HLT-NAACL*. 2013.

[19] Natural Language Toolkit. URL www.nltk.org

[20] Bell, Robert M., Yehuda Koren, and Chris Volinsky. "All together now: A perspective on the netflix prize." *Chance* 23.1 (2010): 24-29.

[21] Herlocker, Jonathan L., Joseph A. Konstan, Al Borchers, and John Riedl. "An algorithmic framework for performing collaborative filtering." *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999.

[22] Montaner, Miquel, Beatriz López, and Josep Lluís De La Rosa. "A taxonomy of recommender agents on the internet." *Artificial intelligence review* 19.4 (2003): 285-330.

[23] Schafer, J. Ben, Joseph Konstan, and John Riedl. "Recommender systems in e-commerce." *Proceedings of the 1st ACM conference on Electronic commerce*. ACM, 1999.

[24] Schein, Andrew I., Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. "Methods and metrics for cold-start recommendations." *Proceedings of the 25th*

annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2002.

[25] Balabanović, Marko, and Yoav Shoham. "Fab: content-based, collaborative recommendation." *Communications of the ACM* 40.3 (1997): 66-72.

[26] Vig, Jesse, Shilad Sen, and J. Riedl. *Computing the tag genome*. Technical report, University of Minnesota, 2010. http://www. grouplens. org/system/files/genome. pdf.

[27] Vig, Jesse, Shilad Sen, and John Riedl. "Navigating the tag genome." *Proceedings of the 16th international conference on Intelligent user interfaces*. ACM, 2011.

[28] Vig, Jesse, Shilad Sen, and John Riedl. "The tag genome: Encoding community knowledge to support novel interaction." *ACM Transactions on Interactive Intelligent Systems (TiiS)* 2.3 (2012): 13.

[29] Adomavicius, Gediminas, and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." *Knowledge and Data Engineering, IEEE Transactions on* 17.6 (2005): 734-749.

[30] About The Music Genome Project. URL https://www.pandora.com/about/mgp

[31] Jinni Taste Based Content Discovery. URL http://www.jinni.com/discovery/

[32] GroupLens Research. URL http://grouplens.org/

[33] MovieLens. URL https://movielens.org/

[34] Sundermeyer, Martin, Ralf Schlüter, and Hermann Ney. "LSTM Neural Networks for Language Modeling." *INTERSPEECH*. 2012.

[35] Mikolov, Tomáš, Stefan Kombrink, Lukáš Burget, Jan Honza Černocký, and Sanjeev Khudanpur. "Extensions of recurrent neural network language model." *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011.

[36] Tsoumakas, Grigorios, and Ioannis Katakis. "Multi-label classification: An overview." *Dept. of Informatics, Aristotle University of Thessaloniki, Greece* (2006).

[37] Zhang, Min-Ling, and Zhi-Hua Zhou. "A review on multi-label learning algorithms." *Knowledge and Data Engineering, IEEE Transactions on* 26.8

(2014): 1819-1837.

[38] scikit-learn LinearSVC. URL http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[39] Deerwester, Scott, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. "Indexing by latent semantic analysis." *Journal of the American society for information science* 41.6 (1990): 391.

[40] Lau, Jey Han, Karl Grieser, David Newman, and Timothy Baldwin. "Automatic labelling of topic models." *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011.

[41] Singhal, Amit. "Modern information retrieval: A brief overview." *IEEE Data Eng. Bull.* 24.4 (2001): 35-43.