Making Use of Affective Features from Media Content Metadata for Better Movie Recommendation Making

John Kalung Leung¹, Igor Griva² and William G. Kennedy³

¹Computational and Data Sciences Department, Computational Sciences and Informatics, College of Science, George Mason University, 4400 University Drive, Fairfax, Virginia 22030, USA
²Department of Mathematical Sciences, MS3F2, Exploratory Hall 4114, George Mason University, 4400 University Drive, Fairfax, Virginia 22030, USA
³Center for Social Complexity, Computational and Data Sciences Department, College of Science, George Mason University, 4400 University Drive, Fairfax, Virginia 22030, USA

{jleung2, igriva, wkennedy}@gmu.edu

Keywords: context-aware, emotion mining, affective computing, recommender systems, movie recommendations, deep learning

Abstract: Our goal in this paper aims to investigate the causality in the decision making of movie recommendations from a Recommender perspective through the behavior of users’ affective moods. We illustrate a method of assigning emotional tags to a movie by auto-detection of the affective attributes in the movie overview. We apply a text-based Emotion Detection and Recognition model, which trained by the short text of tweets, and then transfer the model learning to detect the implicit affective features of a movie from the movie overview. We vectorize the affective movie tags through embedding to represent the mood of the movie. Whereas we vectorize the user’s emotional features by averaging all the watched movie’s vectors, and when incorporated the average ratings from the user rated for all watched movies, we obtain the weighted vector. We apply the distance metrics of these vectors to enhance the movie recommendation making of a Recommender. We demonstrate our work through an SVD based Collaborative Filtering (SVD-CF) Recommender. We found an improved 60% support accuracy in the enhanced top-5 recommendation computed by the active test user distance metrics versus 40% support accuracy in the top-5 recommendation list generated by the SVD-CF Recommender.

1 Introduction

Movie recommendations come from different sources. A more traditional way to make a movie recommendation is by word of mouth through moviegoers who have watched the movie, or relying on elite movie critics who wrote about their opinions of the film, or through news media, publications, and advertisements. Since the dawn of the Internet era in the last century, we rely on machine automation to make movie recommendations using various Recommender methodologies (Bobadilla et al., 2013), (Zhang et al., 2011), (Scheel et al., 2012), and (Kompan and Bielikova, 2014). More recently, we have applied the more advanced Deep Learning (DL) techniques in Recommenders (Zhang et al., 2017). After a century, the field in recommendation making is still in active research (Jannach et al., 2010).

Regardless of the efforts, we have invested in Recommenders research, and they always seem to be more ways to make improvements even in the field (Beel et al., 2015). In this paper, we shall include primary human emotions as an aspect of making movie recommendations through a Recommender (Canales and Martinez-Barco, 2014). Emotion affects human experience and influences our daily activities on all levels of the decision-making process. When a user ponders over a list of recommended items such as songs, books, movies, products, or services, his affective state of preferences influences his decision making on which recommended item he chooses to consume. Emotion plays a role in our decision-making process in preference selection (Naqvi et al., 2006). However, up to now, information retrieval (IR) and Recommender Systems (RS) give little attention to include human emotion as a source of user context (Ho and Tagmouti, 2006). Our goal is to make affective awareness a component of...
We envision that the affective elements of a cinema represented by a low dimension continuous emotion embedding vector denoted as the movie’s emotional vector (mvec). The fact that a film is unique and mutually exclusive from its peers, thus, in theory, a mvec. Some film databases, such as the MovieLens, track users’ movie-watching history, and feedback. Using the moviewatching records of a user, we can formulate a low dimension continuous emotion embedding vector for the user denoted as the user emotional vector (uvec). We obtain a user’s uvec embedding value by taking the average over all the movie’s mvec and the average rating the user has given to all the movies he has watched. Note that, uvec may not be unique. It merely represents the current measure of a user’s affective preference for films that he has enjoyed. The difference between mvec and uvec is that mvec of a movie is static with enduring value throughout its lifetime. Whereas, uvec is dynamic with value change as the user watched and rated movies. The advantage of using the dynamic nature of uvec in making a recommendation is that we are taking the most updated user’s affective preference into consideration of the user decision-making process every time. As the user emotional preference change, so will the movie Recommender to adjust the recommendation making process accordingly. We may be the first party making use of the novelty in leveraging the dynamic nature of uvec over mvec to enhance the process of making movie recommendations of a Recommender. Also, we can leverage mvec to analyze the emotional features of a movie. The continuous embedding nature of the mvec represents the range and strength of moods in a film. In this study, we track six primary human affective features in emotion: “joy”, “sadness”, “hate”, “anger”, “disgust”, and “surprise”. We added “neutral” as the seventh affective feature for convenience in our computing tasks. We normalized the affective features when we compute mvec for a film. Thus, all affective features in mvec will add up to one (1). We can interpret each affective feature in a film as a percentage in decimal. For example, Internet Movie Database (IMDb) is a popular online movie information database that has rated “The Godfather (1972)” as the top movie of all time (IMDb, 2020). Our emotion detector classified the movie with the affective class “hate” and the mvec for the movie depicts in table 1.

Table 1: Affect values of movie: “The Godfather (1972)”.

| Affective Feature | Value          |
|-------------------|----------------|
| neutral           | 0.0840931      |
| joy               | 0.059261046    |
| sadness           | 0.08991193     |
| hate              | 0.23262443     |
| anger             | 0.20177138     |
| disgust           | 0.19720455     |
| surprise          | 0.13513364     |

Glancing through table 1, we perform the affective analysis for the movie in the following way. The movie mvec affective attribute “hate” stands as the dominant feature followed by “anger”, “disgust”, “surprise”, “sadness”, “neutral”, and closed out by “joy”. With the mvec affective features distribution value, we can describe “The Godfather (1972)” movie as a film full of hate. It is an angry movie. The content is disgusting due to the violent nature of the film. However, the story of the movie is full of surprises, but a sad movie. None can say the movie celebrates happiness. By reciting the mvec attributes, we explain the movie. On the same token, we can use the uvec of a specific user who watched “The Godfather (1972)” to explain how well the user has enjoyed the movie. Surprisingly, we can leverage mvec and uvec as an explanation tool in the Recommender recommendation making process.

2 Related Work

Detecting primary human emotion expression in text is a relatively new area of research in Natural Language Processing (NLP). A common approach in identifying the general thought, feeling, or sense in writing is to classify the contextual polarity orientation (positive, neutral, and negative) of opinionated text through the polarity Sentiment Analysis (SA) (Wilson et al., 2005), and (Maas et al., 2011). When applying fine-grained Sentiment Analysis (Fink et al., 2011), researchers can identify the intensity level of the polarity as a multi-
class single-label classification problem (e.g., very positive, positive, neutral, negative, and very negative) [Bhowmick et al., 2009]. However, to determine the mental, emotional state or composure (i.e., mood) in subjective text, Emotional Analysis (EA) can better suit to handle the task [Tripathi et al., 2016]. Here the researcher wants to know the feeling of the writing under examination is in one of the following primary human emotions or moods. The study of primary human emotional expressions started in the era of Aristotle in around 4th century BC [Konstan and Konstan, 2006]. However, not until Charles Darwin (1872 – 1998) revisited the investigation of human emotional expression in the 19th century, which propelled the field to its present stage of modern psychology research [Ekman, 2006]. Paul Ekman et alia in the 1970s developed a Facial Action Coding System (FACS) to carry out a series of research on facial expressions that have identified the following six primary universal human emotions: happiness, sadness, disgust, fear, surprise, and anger [Ekman, 1999]. Ekman later added contempt as the seventh primary human emotion to his list [Ekman et al., 2013]. Robert Plutchik invented the Wheel of Emotions avocated eight primary emotions: anger, anticipation, joy, trust, fear, surprise, sadness, and disgust. Adding to the primary eight emotions are secondary and complementary emotions for a total of 32 emotions depicted on the initial Wheel of Emotions [Plutchik, 2001]. More recent research by Glasgow University in 2014 amended that couple pairs of emotions such as fear and surprise elicited similar facial muscles response, so are disgust and anger. The study broke the raw human emotions down to four fundamental emotions: happiness, sadness, fear/surprise, and disgust/anger [Tayib and Jamaludin, 2016].

In our study, we shall focus our emotion detection work on Ekman’s six primary human emotions. We follow other emotion detection researchers’ footsteps on the same decision, for many have based their work on Ekman’s six primary human emotions [Canales and Martinez-Barco, 2014]. Also, we can make use of the WordNet-Affect, a linguistic resource for a lexical representation of affective knowledge in affective computing on human interaction such as attention, emotions, motivation, pleasure, and entertainment [Valitutti et al., 2004]. It is worthy to note that emotional expression research usually aims at detecting and recognizing emotion types from human facial expression and vocal intonation [De Silva et al., 1997]. However, our EDR study focuses on the mood of text expression instead. The question remains how much of an emotion we can convey through writing.

3 Methodology

Emotion Detection and Recognition on text is a text classification problem, one of the favorite Natural Language Processing and Supervised Machine Learning tasks [Danisman and Alpkocak, 2008]. It is a Supervised Machine Learning task because text classification requires a labeled dataset containing both text documents and associated labels for training the classifier [Medhat et al., 2014]. In the absence of an explicit emotion labeled movie metadata dataset, we build an affective text aware model in two steps, first, through a steadily available domain using tweets data from the Twitter database. Next, we feed the movie text metadata, such as storyline and overviews, to the Emotion Detection and Recognition (EDR) model built from tweets’ affective tags to classify the affective labels for the movie.

3.1 Affective Computing and Machine Learning Modeling

Researchers apply Machine Learning (ML) methodologies to solve two classes of problems: regression and classification [Stone and Veloso, 2000]. For example, to predict tomorrow, we use the stock market index is moving up or down and by how much. It is a regression problem, whereas, to state the polarity or fine-grained sentiment of stock market traders regarding the market performance to come tomorrow, it is a classification problem. In the ML classification, the polarity SA is a multi-class single-label classification problem [Nakov et al., 2019]. However, human emotion usually expresses in a combination of affective moods, such a classification task, a multi-class multi-label classification problem [Li and Ren, 2012]. To simplify our study, we treat the movie metadata text-based emotion detection as a multi-class single-label classification problem instead. Our text-based EDR model’s final step output seven nodes each represents the probability distribution of an affective feature. We then take the argmax function [Gould et al., 2016] from the probability distribution of the candidate nodes as the prediction mood value, as depicted in equation (1):

$$\mathbf{\text{argmax}}_{x \in D} f(x) = \{x | f(x) \geq f(y), \forall y \in D\} \quad (1)$$

The argmax function expresses x as a set of data points for which f(x) obtains the most significant values, if present, of the function.
3.2 Data Preperation

The challenge for our study in emotion detection from movie text-based metadata is to obtain a large enough movie metadata set with mood labels. No such dataset is readily available. We, therefore, need to build the required dataset by deriving it from four different sources. For the movie rating datasets, we obtained the datasets from the MovieLens datasets stored in the GroupLens repository (Harper and Konstan, 2016). We scraped The Movie Database (TMDb) (TMDb, 2018) for movie overviews and other metadata. We derived our emotional word sense set as contextual emotional words synonymous from WordNet (Miller, 1995). Finally, we scraped the Twitter database for tweets with keyword tags that matched our contextual emotion word synonymous (Marres and Weltevrede, 2013). MovieLens contains a “links” file that provides with cross-reference links between MovieLens’ movie id and TMDb’s tmdb id. We connect MovieLens and TMDb datasets through the “links” file.

3.2.1 Extract emotion synonymous from WordNetAffect EmotionLists

WordNet developed an affective knowledge linguistic resource known as WordNet-Affect for lexical representation (Strapparava et al., 2004). The selection and tagging of a subset of synsets convey the emotional meaning of a word in WordNet-Affect. WordNet-Affect emotion lists contain lists of concepts extracted from WordNet-Affect, synsets with six emotions of interest: anger, disgust, hate, joy, sadness, and surprise stored in a compressed file: “WordNetAffectEmotionLists.tar.gz” (Poria et al., 2012). We downloaded the “.gz” file and uncompressed it into six emotion text files. The alternative is to download from GitHub the already uncompressed of the six emotion files from https://github.com/robert-jm/twit-ranker/tree/master/dictionaries/WordNetAffectEmotionLists. Each emotion file contains two columns of information: synsets and the synonymous. Here, the synonymous set of the synset corresponds to an emotion class and store in the corresponding emotion text file. We extract the synonymous column from each emotion text file. We removed duplicate synonymous, sorted the cleansed synonymous, and stored the result in comma-separated values (CSV) format in the corresponding emotion synonymous file. Below in Table 2 contains the statistic of the six emotion synonymous files after performed the data cleansing task.

| mood  | count | synonymous list                                      |
|-------|-------|-----------------------------------------------------|
| anger | 255   | “abhor”, “wrothful”                                  |
| disgust | 53    | “abhorrent”, “yucky”                                 |
| hate  | 147   | “afright”, “unsure”                                  |
| joy   | 400   | “admirable”, “zestfulness”                            |
| sadness | 202   | “aggrieve”, “wretched”                               |
| surprise | 71    | “admiration”, “wondrously”                            |

3.2.2 Extracting tweets from Twitter database

There are many types of tweets on Twitter, a popular social network, and the microblogging platform. In this study, we only work with the regular tweet, 140 characters, or less short message, which posts on Twitter. Almost every user’s tweets are extractable and available to the public. Each tweet is searchable by keyword. We wrote a simple Python script to extract tweets through Twitter’s API (Makice, 2009). We treat each synonymous in an emotion corresponding file as a keyword of a tweet. By looping through all the synonymous in Twitter’s search-by-keyword API, we extract all the tweets with such keyword and store them in a CSV file. The alternative is to extract tweets and store them in a JSON file, as illustrated in (Makice, 2009). For example, if the emotion synonymous is belonging to the anger concept, we will store the retrieved tweet in anger\_raw.csv file. As depicted in Table 3, the anger emotion corresponding file, anger\_syn.txt, has 255 synonymous, we will store all tweets retrieved from the corresponding keywords in anger\_raw.csv.

| mood       | record size |
|------------|-------------|
| neutral    | 19108       |
| joy        | 138019      |
| sadness    | 60381       |
| hate       | 38651       |
| anger      | 17830       |
| disgust    | 19887       |
| surprise   | 15002       |
| unbalance  | 308878      |
| each balanced mood | 15000 |
| each balanced mood train | 12000 |
| each balanced mood test | 3000 |

Table 2: Synonymous statistic of six emotion.

Table 3: Mood datasets gathered from Twitter.
Our mood dataset extracted from Twitter shows an unbalance dataset with “joy” emotion class occupied slightly over a third of the dataset. “sadness” emotion class is half the amount of “joy” while “hate” is half of “sadness”. These three emotion classes, when combined, represent 77% of the emotion dataset. The distribution of the other four emotions ranges from 15,002 to 19,887, more or less evenly distributed. If we apply the dataset directly to Machine Learning modeling without adjustment for the imbalance classes, we will skew our result toward the dominant mood types. We decide to balance the mood dataset by subsampling each affective attribute dataset size to 15,000. We further split each affective dataset into a training dataset with 80% of the samples (12,000) and 20% of the samples for the test dataset (3,000).

Using a brute force method, we scrape the TMDB database for movie metadata, particularly for movie overview or storyline, which contains the subjective writing movie description that we can classify the mood of the text. Knowing we can query the TMDB database by tmdb\_id, a unique movie identifier assigned to a movie. The tmdb\_id starts from 1 and up. However, in the sequence of tmdb\_id, there may be a gap between consecutive numbers. Our effort yields 452,102 records after cleansing the raw data we scraped from TMDB.

### 3.3 Emotion Modeling

Inspired by the recent advance in Natural Language Processing (NLP) for text classification described by Sosa (2017), we develop our text-based EDR model by combining Long Short-Term Memory (LSTM), a variant of Recurrent Neural Network (RNN), and Conv-1D of Convolutional Neural Network (CNN). We build our model similar to the method used in Liu (2020). We define our model architecture consists of two half. The first half is RNN LSTM-CNN Conv-1D architecture, as described in Sosa (2017) that text input process by an LSTM architecture before follow up data processing by a CNN Conv-1D architecture. In contrast, the second half of the model is to reverse the processing order of architecture, CNN Conv-1D-RNN LSTM. Input first process by a CNN Conv-1D architecture before feeding it to an RNN LSTM architecture. The two half of the architecture then combine to feed data into a max-pooling layer of a CNN for a pooling operation to select the dominant feature in the filter’s regional feature map. Next, data passes into a flattening layer of a CNN to convert data into a one-dimensional array before feed data to a fully connected dense layer of a CNN. The dense layer’s output will feed to a set of nodes that are equal to the number of classes the architecture aims to classify. Each of the output nodes holds the output distribution value of its class. In the final act, a softmax activation function examine and activate the appropriate class node accordingly.

We use bi-directional RNN LSTM and CNN Conv1D architectures to build our model. In the first half of the model, the RNN LSTM-CNN Conv-1D phase, we use two bi-directional LSTM for the RNN LSTM architecture and seven Conv1D of the CNN architecture. In the second half of the model, the CNN-LSTM phase, we apply seven pairs of Conv1D of the CNN architecture and two bi-directional LSTM for the LSTM architecture. Follow the idea illustrated in [Kim, 2014], we prepared two identical input layers of embedding matrix constructed from a pre-trained GloVe embedding matrix similar to [Pennington et al., 2014]. We build the two input layers of the embedding matrix with one of the input embedding layers set to “trainable”, while the other is not, i.e. “frozen”. During the processing of the first half of the model, the RNN LSTM-CNN Conv-1D phase, the “trainable” input layer occupies one of the bi-directional LSTM architecture. In contrast, the “frozen” input layer fills the other.

Similarly, when processing the second half of the model, the CNN Conv-1D - RNN LSTM phase, a “trainable” input layer, and a “frozen” input layer will occupy each pair of the Conv1D units. We obtain 55.6% accuracy in classifying the emotion class of the tweets’ balanced dataset, as depicted in table 4 and the confusion matrix in figure depicts the performance of the seven emotion classifier. The classification result is acceptable to serve our purpose since our goal is not to build the best emotion text classifier, but a usable one to classify the emotion class of movie overviews.

| emotion     | precision | recall | f1-score | support |
|-------------|-----------|--------|----------|---------|
| neutral     | 0.47      | 0.77   | 0.59     | 2992    |
| joy         | 0.63      | 0.53   | 0.58     | 3030    |
| sadness     | 0.64      | 0.44   | 0.52     | 3034    |
| hate        | 0.64      | 0.51   | 0.57     | 2933    |
| anger       | 0.62      | 0.08   | 0.35     | 2984    |
| disgust     | 0.44      | 0.45   | 0.44     | 2987    |
| surprise    | 0.35      | 0.51   | 0.43     | 3040    |
| accuracy    |           |        |          |         |
| macro avg   | 0.57      | 0.56   | 0.55     | 21000   |
| weighted avg| 0.57      | 0.56   | 0.55     | 21000   |
For comparison purposes, we include the ERD performance result on unbalanced mood tweets data listed in table 5. We also depicted the confusion matrix of the seven emotion unbalanced tweets dataset in table 2.

Table 5: EDR performance on unbalanced mood tweets dataset.

|     | precision | recall | f1-score | support |
|-----|-----------|--------|----------|---------|
| neutral | 0.34     | 0.50   | 0.41     | 3758      |
| joy   | 0.73     | 0.85   | 0.78     | 27719      |
| sadness | 0.57     | 0.66   | 0.61     | 12126      |
| hate  | 0.68     | 0.51   | 0.58     | 7710       |
| anger | 0.74     | 0.46   | 0.56     | 3518       |
| disgust | 0.49     | 0.22   | 0.30     | 3972       |
| surprise | 0.70     | 0.24   | 0.36     | 2972       |
| accuracy | 0.65     |        |          | 61775      |
| macro avg | 0.61     | 0.49   | 0.51     | 61775      |
| weighted avg | 0.65     | 0.65   | 0.64     | 61775      |

Let us revisit the example we gave in the mood analysis of the most top movie of all time rated by IMDb, “The Godfather (1972)”. We built another version of our emotion detection classifier model using the unbalanced mood dataset. The classifier classified “hate” is the most dominant mood feature for the movie, and the mvect for the film with the model built from the unbalanced dataset depicts in 6. The mvect depicts the mood attributes descending order of the movie is “hate”, “anger”, “joy”, “sadness”, “disgust”, “neutral”, and “surprise”. Comparing to table 1 which shows the mood attributes descending order as “hate”, “anger”, “disgust”, “surprise”, “sadness”, “neutral”, and “joy”. Any person who has watched “The Godfather (1972)” movie would probably favor the mood analysis result done by the emotion detection classifier model using the balanced mood dataset.

Table 6: Affect values of movie: “The Godfather (1972)” derived from unbalanced mood dataset.

|     |     |     |     |     |
|-----|-----|-----|-----|-----|
| neutral | 0.04276474 |     |     |     |
| joy |     |     |     |     |
| sadness |     |     |     |     |
| hate |     |     |     |     |
| anger |     |     |     |     |
| disgust |     |     |     |     |
| surprise |     |     |     |     |

3.4 Emotion Prediction

We build a seven text-based emotion predictor for movie overviews from the seven emotion tweet classifier model. We run the predictor through all the 452, 102 overviews scraped from the TMDb database to generate a TMDb movie overview with an emotion label dataset. As mentioned before, MovieLens datasets come in different sizes. We will work with the following MovieLens datasets: ml-1m,
which contains about one million rating information of movies; ml-20m dataset, 20 million rating information; ml-latest-small dataset, about ten thousand rating information of 610 users; ml-latest-full dataset, holds 27 million rating information; and the recently leased ml-25m dataset, with 25 million rating information. (Note the name of the MovieLens dataset conveys the number of ratings, movies, users, and tags contained in the dataset.) Table 7 depicts the number of ratings, users, and movies; each of the MovieLens datasets contain. In each of the depicted MovieLens dataset, it provides a links file to cross-reference between MovieLens and two other movie databases, TMDb and IMDb, through movie_id, tmdb_id, and imdb_id. MovieLens maintains a small number of data fields, but users can link to TMDb and IMDb databases via the links file to access other metadata that MovieLens is lacking.

The ml-latest-full datasets maintain the most significant number of movies in MovieLens dataset collection. However, the ml-latest dataset will change over time and is not a proper use for reporting research results. We use the ml-latest-small, and ml-latest-full datasets in proof of concept and prototyping, not research reporting work. The other MovieLens 1M, 20M, and 25M datasets are stable benchmark datasets which we will use for research reporting work.

Although we have scraped 452,102 movie overviews from TMDb when merging with MovieLens, we only make use of one-eighth of the number of overviews we have collected. Showing in Table 8 is the number of movie overviews the MovieLens datasets can extract from TMDb after performing the data cleaning task.

### Table 7: MovieLens datasets.
| dataset         | ratings | users    | movies    |
|-----------------|---------|----------|-----------|
| ml-1m           | 1M      | 6000     | 4000      |
| ml-20m          | 20M     | 138000   | 27000     |
| ml-25            | 25M     | 162000   | 62000     |
| ml-latest-small | 100K    | 600      | 9000      |
| ml-latest-full  | 27M     | 280K     | 58000     |

### Table 8: Number of overview in MovieLens extracted from TMDb.
| dataset          | number of overviews |
|------------------|---------------------|
| ml-1m            | 1M                  |
| ml-20m           | 26603               |
| ml-25            | 25M                 |
| ml-latest-small  | 9625                |
| ml-latest-full   | 56314               |

4 Implementation

4.1 Uvec and Mvec

For our study, we need a movie Recommender System to evaluate the performance of uvec and mvec. We envision uvec and mvec play a role in the tail end of making movie Recommendations, i.e., during the stage of the top-N movie recommendation making process. Any movie Recommenders can fit to support the evaluation of uvec and mvec. In our case, we develop a Collaborative Filtering-based movie Recommender System (CFRS) based on the SVD algorithm. We then added functions to support uvec and mvec operations in the enhancement of making movie recommendations. We start by adding functions in the SVD-CFRS to support the loading, storing, extraction, and manipulation operations of uvec and mvec. The added uvec and mvec support functions do not interfere with any normal RS operation, including the ordinary making movie recommendations process. We let our SVD-CFRS run in its usual way serving routine recommendation requests: no uvec and mvec involvement in the movie recommendation making just yet. Before we start to evaluate uvec and mvec, we prepare each user has the uvec by computing the average of all mvec of the movies the user has watched. We deployed the MovieLens ml-latest-small dataset to be the test set and randomly pick a user, user_id 400, as the active test user.

Our active test user’s uvec depicted in Table 9 representing the overall average of 43 movies’ mvec the user_id 400 has watched. We ask the Recommender to make movie recommendations for the user_id 400 in a business-as-usual way. The Recommender furnishes a top-N, where N is 20, movie recommendation list for the user_id 400 as depicted in Table 11. Using the uvec of the user_id 400, we compute the pairwise similarity between the user_id 400 and each recommended movie’s mvec on the top-N list. We sorted the pairwise distance metrics top-N list computed using uvec in the descending order to match up the presentation ordering of the Recommender’s top-N list. We applied five different distance metrics for computing the uvec’s pairwise distance metrics top-N list. We depicted the comparative top-N results in Table 12. The five distance metrics we employed in the comparison were Euclidean distance, Manhattan distance, Minkowski distance, Cosine similarity, and Pearson correlation with their formula illustrated in Equation 2 through 7.

\[
\text{Euclidean distance} (x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2}
\]
Manhattan distance \((x, y) = \sum_{i=1}^{n} |x_i - y_i|\)

(3)

Minkowski distance \((x, y) = \left( \sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}\)

(4)

Inner product \((x, y) = \sum_{i=1}^{n} x_i y_i\)

(5)

Cosine similarity \(\text{CosSim}(x, y) = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2} \sqrt{\sum y_i^2}}\)

(6)

Pearson correlation \(\text{PearCorr}(x, y) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2} \sqrt{\sum (y_i - \bar{y})^2}} = \frac{\text{CosSim}(x - \bar{x}, y - \bar{y})}{||x - \bar{x}|| ||y - \bar{y}||}\)

(7)

where \((x, y)\) are vectors \(x = (x_1, x_2, \cdots, x_n)\) and \(y = (y_1, y_2, \cdots, y_n)\).

5 Evaluation

5.1 Findings

The following table 9 depicts the average mood value of user id 400, while table 10 shows the weighted average mood value of the active test user.

Table 9: The average mood value of user id 400 user.

|          | neutral | joy     | sadness | hate  |
|----------|---------|---------|---------|-------|
|          | 0.16352993 | 0.08873525 | 0.08006808 | 0.11933819 |

Table 10: The weighted average mood value of user id 400 user.

|          | neutral | joy     | sadness | hate  |
|----------|---------|---------|---------|-------|
|          | 0.14755723 | 0.15938389 | 0.14330099 | 0.123581309 |

The following table 11 depicts the top-N movie recommendation list generated by our SVD-CF Recommender for the active test user id 400.

Table 11: TopN recommendation list generated by SVD CF recommender for user id 400 user.

| no. | mid | title |
|-----|-----|-------|
| 0   | 527 | Schindler’s List (1993) |
| 1   | 5952 | Lord of the Rings: The Two Towers, The (2002) |
| 2   | 2858 | American Beauty (1999) |
| 3   | 2329 | American History X (1998) |
| 4   | 2028 | Saving Private Ryan (1998) |
| 5   | 1089 | Reservoir Dogs (1992) |
| 6   | 110 | Braveheart (1995) |
| 7   | 1291 | Indiana Jones and the Last Crusade (1989) |
| 8   | 4226 | Memento (2000) |
| 9   | 91529 | Dark Knight Rises, The (2012) |
| 10  | 68157 | Inglourious Basterds (2009) |
| 11  | 6016 | City of God (Cidade de Deus) (2002) |
| 12  | 589 | Terminator 2: Judgment Day (1991) |
| 13  | 1704 | Good Will Hunting (1997) |
| 14  | 1200 | Aliens (1986) |
| 15  | 1214 | Alien (1979) |
| 16  | 1 | Toy Story (1995) |
| 17  | 99114 | Django Unchained (2012) |
| 18  | 7361 | Eternal Sunshine of The (2012) |
| 19  | 1136 | Monty Python and the Holy Grail (1975) |

The following table 12 depicts the comparison of the top-N recommendation list generated by the five distance metrics for user id 400 using the weighted mood values for the uvec.

The following table 13 depicts the comparison of the top-N recommendation list generated by the five distance metrics for the active test user id 400 using the weighted mood values for the uvec. To get the average weighted uvec for the active test user id 400, we multiply the averaged user id 400 uvec with the average normalized rating value of the watched movie. We depicted the average weighted uvec of user id 400 in table 10. Using the weighted uvec of user id 400, we compute the five weighted distance metrics de-
Table 12: Comparison of the top-N recommendation list generated by the five distance metrics for user_id 400 user.

| no. | mid | Euc   | Man   | Min   | Cos   | Pear  |
|-----|-----|-------|-------|-------|-------|-------|
| 0   | 527 | 1291  | 1291  | 1291  | 91529 | 110   |
| 1   | 5952| 4226  | 5952  | 4226  | 68157 | 2329  |
| 2   | 2858| 5952  | 4226  | 7361  | 527   | 91529 |
| 3   | 2329| 7361  | 1089  | 5952  | 2329  | 527   |
| 4   | 2028| 1089  | 7361  | 1089  | 2858  | 2028  |
| 5   | 1089| 1200  | 2028  | 1200  | 6016  | 68157 |
| 6   | 110 | 2028  | 589   | 1704  | 110   | 1214  |
| 7   | 1291| 1704  | 1704  | 2028  | 1     | 1     |
| 8   | 4226| 589   | 1200  | 589   | 99114 | 1200  |
| 9   | 91529| 1214  | 1214  | 1136  | 1089  |
| 10  | 68157| 99114 | 1     | 1136  | 1214  | 6016  |
| 11  | 6016| 110   | 1136  | 99114 | 2028  | 2858  |
| 12  | 589 | 1136  | 110   | 110   | 1      | 91529 |
| 13  | 1704| 1     | 2329  | 1     | 1200  | 1136  |
| 14  | 1200| 6016  | 527   | 2858  | 1704  | 1704  |
| 15  | 1214| 2329  | 99114 | 6016  | 1089  | 589   |
| 16  | 1   | 527   | 6016  | 527   | 7361  | 5952  |
| 17  | 99114| 2858  | 91529 | 2329  | 5952  | 7361  |
| 18  | 7361| 68157 | 2858  | 68157 | 4226  | 1291  |
| 19  | 1136| 91529 | 68157 | 91529 | 1291  | 4226  |

Table 13: Comparison of the top-N recommendation list generated by the five distance metrics for user_id 400 user using weighted uvec.

| no. | mid | wEuc  | wMan  | wMin  | wCos | wPear |
|-----|-----|-------|-------|-------|------|-------|
| 0   | 527 | 1291  | 1291  | 1291  | 91529| 110   |
| 1   | 5952| 4226  | 5952  | 4226  | 68157| 2329  |
| 2   | 2858| 5952  | 4226  | 7361  | 527  | 91529 |
| 3   | 2329| 7361  | 1089  | 5952  | 2329  | 527   |
| 4   | 2028| 1089  | 7361  | 1089  | 2858  | 2028  |
| 5   | 1089| 1200  | 2028  | 1200  | 6016  | 68157 |
| 6   | 110 | 2028  | 589   | 1704  | 110   | 1     |
| 7   | 1291| 1704  | 1704  | 2028  | 1     | 1     |
| 8   | 4226| 589   | 1200  | 589   | 99114| 1200  |
| 9   | 91529| 1214  | 1214  | 1136  | 1089  |
| 10  | 68157| 99114 | 1     | 1136  | 1214  | 6016  |
| 11  | 6016| 110   | 1136  | 99114 | 2028  | 2858  |
| 12  | 589 | 1136  | 110   | 110   | 1      | 91529 |
| 13  | 1704| 1     | 2329  | 1     | 1200  | 1136  |
| 14  | 1200| 6016  | 527   | 2858  | 1704  | 1704  |
| 15  | 1214| 2329  | 99114 | 6016  | 1089  | 589   |
| 16  | 1   | 527   | 6016  | 527   | 7361  | 5952  |
| 17  | 99114| 2858  | 91529 | 2329  | 5952  | 7361  |
| 18  | 7361| 68157 | 2858  | 68157 | 4226  | 1291  |
| 19  | 1136| 68157 | 68157 | 91529 | 1291  | 4226  |

5.2 Comparison of Top-N made by Distance Metrics

We combined the following rating datasets, except the ml-latest-small from the MovieLens into one dataset by merging ml-1m, ml-20m, ml-25m, and ml-latest-full datasets. We extracted all data points of user_id 400 as the active test user under examination from the combined rating dataset. We removed duplicated data points based on the movie_id and timestamp from the combined rating dataset that matched the data points in user_id 400. The combined rating dataset employs here as movies to-be-watched by the active test user_id 400 in some future timeline. We employed ml-latest-small as the input dataset for our SVD-CF Recommender to get the top-N recommendation list for our active test user_id 400. The SVD-CF Recommender is never aware of the combined user_id 400 data points except those contained in the ml-latest-small dataset. Once we excluded all the duplicated data points found in ml-latest-small from the combined rating datasets, we have 209 user_id 400 data points for validation work. We extracted data points from the combined dataset, which matched the top-N recommendation generated by the Recommender, as depicted in Table 14. We put the extracted data points into two groups: data points found in the top-N and not found in the top-N. There are 8 data points in the combined dataset sorted by timestamp in ascending order to reflect the order of the movie the active user has watched. That is a good sign because it shows the SVD-CF Recommender works at a 40% supporting rate, eight out of twenty recommendations coincided with the preference of the active test user. If we draw a cutoff point at top-5 across the comparison of the top-N recommendation list, the following four movie_id data points in the combined dataset matched: 1291, 4226, 7361, and 2028. Cosine takes fifth place; it only hit the 4th spot in the top-5 list. Pearson takes the fourth place and hit the top second and fifth spots on the top-5 list. Manhattan takes third place, hitting the top 1, 3, and 5. Euclidean takes second place, hitting the top 1, 2, and 4. Minkowski wins the round by hitting the top 1, 2, and 3 spots.

6 Future Work

We plan to elaborate our affective computing study by building an Emotion Aware Recommender using the emotion labeled tags obtained from movie overviews through the Tweets Affective Classifier. We also plan
Table 14: Common data points found in TopN recommendation list and in active user user_id 400.

| no. | movie_id in topN | movie_id not in topN |
|-----|------------------|----------------------|
| 1   | 1                | 110                  |
| 2   | 2329             | 527                  |
| 3   | 1291             | 1089                 |
| 4   | 589              | 1136                 |
| 5   | 1704             | 1200                 |
| 6   | 2028             | 1214                 |
| 7   | 4226             | 2858                 |
| 8   | 7361             | 5952                 |
| 9   |                  | 6016                 |
| 10  | 68157            |                      |
| 11  | 91529            |                      |
| 12  |                  | 99114                |

to make use of affective features in users’ emotion profiles to enhance Group Recommender in group formation, group dynamic, and group decision making. In our study, we would like to find better metrics to measure the performance of affective computing.

7 Conclusion

In this paper, we illustrate a strategy to generate affective features for movies by transfer learning techniques utilizing a different domain Emotion Detection and Recognition (EDR) model classifier. We developed the EDR model to detect and recognize seven emotional features in tweets through affective tags stored in the Twitter database. We then transfer the learning of the EDR model from classifying the emotional features of tweets to predict the moods of a movie through the movie description in the movie overview. We scraped the TMDb database for movie overviews and metadata. Through the EDR program, we generate emotional features, mvec, for each collected movie from TMDb. We gather movie datasets that contain rating information from the MovieLens repository. We use the rating dataset of MovieLens to build an SVD-CF Recommender. We add functions to support uvec and mvec in enhancing the Recommender in generating the top-N movie recommendations. We randomly pick an active user, user_id 400, as the test candidate. We generate a top-N with N set to 20 to generate movie recommendations through the SVD-CF Recommender. We compute the uvec for the user_id 400 by average all the mvec of movies that the testing candidate has watched. We calculate the Euclidean, Manhattan, Minkowski, Cosine, and Pearson distance metrics and compare the five distance metrics’ rankings against the Recommender top-N. We found Minkowski distance metrics performed the best at 60% support accuracy versus the 40% accuracy made by the Recommender. A 20% improvement in top-N movie recommendations. We also make the following observations:

- Text-based NLP EDR modeling technique works and can apply to solve a real-world problem where the abundance of subjective writing is available.
- Text-based EDR model is transferable from one domain to another, and all it requires is that the target text is in subjective writing form.

REFERENCES

Beel, J., Gipp, B., Langer, S., and Breitinger, C. (2015). Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, pages 1–34.

Bhowmick, P. K., Basu, A., Mitra, P., and Prasad, A. (2009). Multi-label text classification approach for sentence level news emotion analysis. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 261–266. Springer.

Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46:109–132.

Canales, L. and Martínez-Barco, P. (2014). Emotion detection from text: A survey. In *Proceedings of the Workshop on Natural Language Processing in the 5th Information Systems Research Working Days (JISIC)*, pages 37–43.

Danisman, T. and Alpkocak, A. (2008). Feeler: Emotion classification of text using vector space model. In *AISB 2008 Convention Communication, Interaction and Social Intelligence*, volume 1, page 53.

De Silva, L. C., Miyasato, T., and Nakatsu, R. (1997). Facial emotion recognition using multi-modal information. In *Proceedings of ICICS, 1997 International Conference on Information, Communications and Signal Processing. Theme: Trends in Information Systems Engineering and Wireless Multimedia Communications* (Cat., volume 1, pages 397–401. IEEE.

Ekman, P. (1999). Basic emotions. *Handbook of cognition and emotion*, 98(45-60):16.

Ekman, P. (2006). *Darwin and facial expression: A century of research in review*. Ishik.

Ekman, P., Friesen, W. V., and Ellsworth, P. (2013). *Emotion in the human face: Guidelines for research and an integration of findings*, volume 11. Elsevier.

Fink, C. R., Chou, D. S., Kopecky, J. J., and Llorens, A. J. (2011). Coarse- and fine-grained sentiment analysis of social media text. *Johns hopkins apl technical digest*, 30(1):22–30.

Gould, S., Fernardo, B., Cherián, A., Anderson, P., Cruz, R. S., and Guo, E. (2016). On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*. 

Danisman, T. and Alpkocak, A. (2008). Feeler: Emotion classification of text using vector space model. In *AISB 2008 Convention Communication, Interaction and Social Intelligence*, volume 1, page 53.
