Visualizing the Emotion Flow in Video on a Timeline by Predicting Emotions in Subtitles

Tejas Dharamsi, Siddharth Varia, Nitesh Surtani and **Keerti Agrawal**
Data Science Institute, Columbia University
Broadway W 120th St, New York, NY 10027
{td2520,sv2504,ns3148,ka2601}@columbia.edu

Abstract

Traditionally, movies are manually tagged based on their genre namely comedy, thriller, horror, romance etc. attributing to the overall theme of the movie. In this paper, we attempt to automatically classify emotions in the movie at the scene level, thereby providing more detailed insight to the viewer. We adopt a supervised classification framework for the task by using the parallelly aligned subtitles of the movie manually annotated with corresponding emotions. The flow of emotions is ultimately visualized on the movie timeline, which represents the variation of each emotion as the movie progresses. The paper explores various novel features for predicting the emotions from movie dialogues and experiments with the classification models like Logistic Regression and SVM at different hyperparameter settings. The feature engineering reveals that movie behaves like a time series data, following the trend of the previous scene emotions, with certain emotion transition points and, thus, giving a huge boost to our classification model. Our system highly outperforms the models previously tested on this data achieving an overall accuracy of 71.24% and f-score of 0.71. The task has applications in searching based on emotions, automatic genre prediction of movies, various comparative study of movies and automatic summarization of movies.

1 Introduction

An emotion is a particular feeling that characterizes a state of mind, such as joy, anger, love, fear and so on. Emotions are expressed in videos in a variety of ways, among which the prominent sources are: Dialogue Content, Facial Expressions, Music and Background Scene. In this paper, we attempt to detect emotions in the dialogues of the video which are embedded textually in the subtitles file. The interesting part is that the subtitles are parallelly aligned to the scenes in the movie, and thus, predicting the emotion in the subtitle text in turn classifies the emotion of the scene. Ultimately, we map the flow of emotions in a movie on its timeline, which gives quite interesting insights on the change of emotions as the movie progresses. This serves many purposes, including:

1. **Search:** Allowing search based on emotions. For example, retrieving the most fearful scenes from an horror movie like Shutter or the comedy scenes from Silicon Valley.

2. **Comparative analysis of movie:** It is quite interesting to compare the pattern of emotion flow in movies of different genre, answer question likes: Is movie A more funny than movie B?, compare the movies from 80s vs the movies of 2000 of same genre to find whether the emotion variation quite fast today? etc.

3. **Summarization:** For example, automatically generating summaries that capture the different emotional states of the characters in a movie.

With increasing competition in subscription based services like netflix, hulu and amazon prime, it will become imperative for these services to build enhanced video information retrieval and recommendation systems. Emotion detection from movie or video subtitles can aid both video information re-

trieval and recommendation systems. In this paper, we propose to build a system that can automatically visualize the flow of different emotions in movies/videos on a timeline. Finding the highlights of a video helps to get a quick review in advance or to watch the major scenes once again, and can also be extended for genre classification of the movie. To achieve this, we will develop an efficient emotion classification model from subtitles of the movie/video, which will predict the occurrence of each of the 6 emotions (proposed by Ekman [5]: anger, disgust, fear, happy, sad, surprise or emotionless) over short equally-sized interval periods.

In this paper, we focus on 3 main sub-tasks: (i) Developing an high performance emotion classification model for movie domain; (ii) Visualizing the emotions in a movie on the timeline; and (iii) Making various comparisons of different kind of movies and gaining useful insights from the data. Task (i) consists of two important sub-tasks: (a) Exploring various features that are relevant for emotion classification in movies and performing an feature additive study to find the significance of each feature; and (b) Exploring various supervised techniques to model this problem. This will include experimentation with different ML models and perform cross-validation experimentation, parameter tuning using grid search etc. This is discussed in more detail in the subsection below. In task (ii), once we have an emotion classifier in place, we will run it for each movie to visualize and plot the emotions of each scene on a timeline to see the flow of emotions. This module will gain insights from Mohammad (2012) work. In task (iii), we will draw useful comparisons from the timeline of different movies, gaining some insights from Mohammad (2012) work.

The paper is organized as follows: Section 2 (**Related Works**) describes previous works on Emotion detection from text; Section 3 (**Dataset and Lexicon**) describes the emotion lexicon used for the task and the dataset used for experiments, Section 4 (**Feature Selection**) and Section 5 (**Classification Experiments**) discusses, experiments and provide insights on the features used for classification and various classification models tried out for the task. Section 6 (**Results**) discusses the performance of the models and presents the visualization of the emotion flow of the movie timeline. Section 7 concludes the paper and lays the path for the future direction.

2 Related Work

Over the last decade, there has been considerable work done on similar grounds of identifying emotions in movie scenes using subtitles (Chetan and Kim, 2009). They have used unsupervised learning approach to detect and clip emotional scenes of 8 movies with scene level emotion annotation, having a total dataset size of 8000 sentences. Our selection of features has been highly inspired from this work, dependency features, punctuations, emotion polarity of last k sentences etc. Another approach by Agrawal et al. (2012) to detect emotions from text uses NAVA (Noun Adjective Verb Adverb) as emotion words, dependency parsing for structure level information and stemming of the words. Strapparava et al. (2008) set out to classify the news headlines into Ekmans 6 emotion categories. Their approaches vary from just looking for emotion words in the headlines (baseline), using LSA to compute similarities between headlines and emotion classes and using blog posts to train a naive bayes classifier later used to classify the news headlines. Their baseline approach yields best precision whereas their LSA-all words approach yields best recall and best F-measure overall.

The paper Shelke (2014) discusses advantages and limitations of various approaches to emotion detection in text data. One of the approaches mentioned is the keyword based (bootstrapping) approach wherein you use synonyms and antonyms from wordnet to determine semantic orientation of the adjective. If there are enough adjective with known orientation in seed then one can determine orientation of all adjective. The problems with this method were 1) dealing with word sense disambiguation 2) inability to detect words without keywords and 3) doesnt provide enough linguistic information. These findings motivate us to avoid seed based approach.

The survey paper Canales et al(2014) focuses on studying recent work on lexical and machine learning approaches and these works are classified in accordance with the emotional model and the approach level. The paper concludes that the machine learning approaches are better option for detecting emo-

tions in texts. Although, it is important to use a good lexical resource as features in machine learning algorithms to obtain good results. This paper basically motivates us to use machine learning approach to detect emotions in the subtitles in our project. The blog from Microsoft discusses and introduces the challenges in emotion detection and authors advent while solving this problem using a deep learning based approach. Sinha et al. (2015) use word2vec to obtain word embeddings from Wikipedia corpus and use the average of emotion word vectors as the vector for each dialogue. These dialog vectors are used to train various classifiers.

Closest to our task is the work of Park et al. (2011) where authors annotate movie dialogs based on emotion in their text. Their approach is also based on some kind of similarity measure. They start with 3 concepts/words 'emotional state', 'emotion', 'feeling' and expand their concepts by exploiting the semantic relation hyponyms in wordnet to obtain 43 emotional concepts, later reducing it to 30. However we think that their approach has few weaknesses, specifically for few words, they could not compute distances as there was no path between the concept word and the given word. Their system also could not handle polysemy and emotion expressed in idioms. Our work differs from this mainly in avoiding wordnet as a dependency.

3 Lexicon and Datasets

3.1 Lexicon

We have used the Ekmans emotion schema for our task with 6 primary emotions: happiness, sadness, anger, fear, disgust and surprise. The dataset we use for our experiments is annotated with one of these 6 emotions, or emotionless if no emotion is present in the scene. There are various emotions lexicon developed in the past depending on the task, with each having its pros and cons. We use the EmoLex lexicon (Figure 1), which is a manually created lexicon of 14,182 words with 25,000 word senses markings, encoding eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). We map joy to happiness and do not make use of the trust and anticipation lexicon to make this lexicon consistent with the Ekmans lexicon.

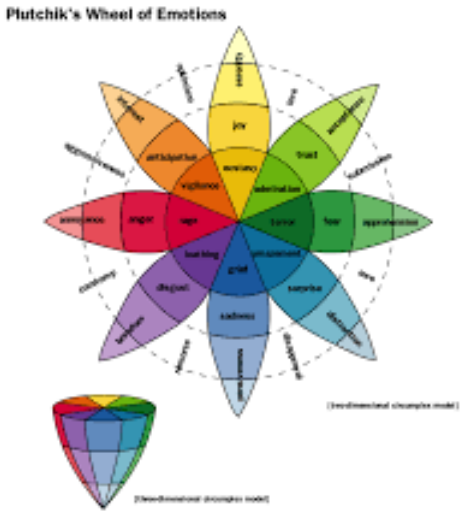

Figure 1: Emolex Emotion Schema

3.2 Datasets

We work with the emotion labeled dataset from Sinha et al. (2015). The dataset consists of the subtitles of 3 videos (i.e. Friends episode The one where everyone finds out, Walking dead episode Days gone bye and the movie Titanic) manually tagged with one of the 6 emotions Ekmans emotion, or emotionless if none of the emotions exist. The annotation is done at the scene level, as recorded in the subtitle, which is generally an equivalent of a sentence. The dataset consists of 2046, 243, 471 scene annotations from Titanic, Friends and Walking dead respectively. The distribution of each relation in these datasets is presented below in Table 1.

The above class distributions have baselines of 30.8%, 51.8% and 43.52% for Titanic, Friends and Walking Dead respectively, with emotionless class being the majority in each. Since the dataset size is relatively small for learning a classification model, we all conduct experiments on the combined dataset of 3 datasets, which yields a baseline of 35%. To overcome the issue of small dataset, we perform k-fold cross validation experiments, to get the performance estimate on the overall dataset, which will be discussed in the following sections.

	Titanic		Friends		Walking Dead		Combined	
Class	**Count**	**(%)**	**Count**	**(%)**	**Count**	**(%)**	**Count**	**(%)**
Anger	176	8	11	4.5	52	11.04	239	9
Disgust	211	10.3	3	1.23	9	1.9	223	8
Emotionless	631	30.8	126	51.8	205	43.52	962	35
Fear	217	10.6	8	3.29	49	10.4	274	10
Happy	317	15.5	61	25.1	36	7.6	414	15
Sad	271	13.24	19	7.8	80	17	370	13
Surprise	223	10.89	15	6.17	40	8.49	278	10
Total	2046	100.0	243	100.0	471	100	2760	100.0

Table 1: Distribution of emotions in Datasets

4 Feature Selection

We have explored various novel features for the task of predicting the emotion of the scene and performed feature engineering to find out which features are actually significant in predicting the emotions in movies. One very crucial insight we gained from our experiments is that movie behaves like a time series data and thus, there are emotion intervals until they are switched by another emotion. Apart from trying the generally used Unigrams and Bigrams features in the model (considered as our baselines), we used features like:

1. **Cumulative Emotion in Sentence:** This is a feature vector of seven emotions and: 6 Ekmans emotion and one emotionless, and two features for the positive and negative sentiment of the dialogue. Each feature consists of a value between 0 and 1. The cumulative emotion of a given dialogue is computed as:

   For each emotion j:

   $$Score[j] = \sum_{i \in W} emotion\_score(w_{ij}) \quad (1)$$

2. **POS-Percent (JJ, NN, RB, VB):** This comprises of a feature vector of size 4 consisting of the percentage of POS tags in the given dialogue normalized between 0 and 1. This gives us an insight that a dialogue with more adjectives and adverbs is more likely to be having an emotion. The cumulative sentiment of the NAVA (Noun, Adjective, Verb, Adverb) tells whether the dialogue has a positive or a negative emotion.

3. **Last k-tags (Time Series):** This is a feature vector of size k, where k is the number of previous predicted labels used for predicting the current label. In this article we use tags and labels interchangeably.

4. **Punctuations:** This comprises of three binary features: a feature for exclamation, question and musical notations.

To extract these features, we performed the preprocessing of text. First, we tokenized the text to split into words, POS tagged those words to extract Nouns (NN), Verbs (VB), Adjective (JJ), Adverbs (RB) words and then Lemmatized the words to use only their root form to reduce the morphological variants.

5 Classification Experiments

5.1 Classification Models

The problem of predicting emotions from dialogues in a movie is formulated as supervised learning task using learning algorithms such as Logistic regression (LR), Linear SVM (LSVM) and Random Forest (RF). Sinha et al.(2015) used SVM and Random Forest Models for their task which motivated us to use the same apart from LR to compare the results. The implementation of these models was done with the help of Pythons Scikit Learn Machine Learning Toolkit. For LR, we used the default optimizer 'liblinear' along with L2 regularization. For SVM, we tried linear and RBF kernels along with L2 regularization and squared hinge loss. However the SVM results reported here are for linear kernel as it yielded better results than RBF kernel. For Random

	Titanic			Friends			Walking Dead			Combined		
Class	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
happy	0.79	0.79	0.79	0.60	0.60	0.60	0.53	0.44	0.48	0.75	0.75	0.75
surprise	0.62	0.62	0.62	0.00	0.00	0.00	0.20	0.05	0.08	0.56	0.55	0.55
sad	0.76	0.76	0.76	0.12	0.05	0.07	0.63	0.60	0.62	0.70	0.71	0.70
disgust	0.59	0.55	0.57	0.00	0.00	0.00	0.00	0.00	0.00	0.58	0.51	0.54
anger	0.66	0.66	0.66	0.00	0.00	0.00	0.53	0.53	0.53	0.62	0.62	0.62
fear	0.80	0.80	0.80	0.33	0.12	0.18	0.65	0.57	0.61	0.76	0.78	0.77
emotionless	0.72	0.73	0.73	0.64	0.86	0.73	0.61	0.77	0.68	0.71	0.73	0.72
F_{avg}			0.71			0.23			0.43			0.66

Table 2: Performance on different datasets using Logistic Regression model

Forest, we set the number of trees equal to 10 and used 'Gini' as the split criteria.

5.2 Experiments

Through our experiments we aim to determine the extent to which we can automatically classify a dominating emotion within a sentence and which features contribute immensely to serve this purpose. We run all three previously mentioned on the available three dataset namely Titanic, Friends, Walking Dead and a dataset assimilated by combining by all of the three mentioned above, hereafter referred as combined. We run these experiment twice, once including the emotionless labels and other one without. We use all the features described in the feature selection section for these experiments. The baseline for these experiment was considered as the proportion of the most dominant emotion in the dataset i.e. emotionless in the original three dataset. Since the size of dataset available to run these experiment were really small we decided to perform 5-fold cross validation experiments. Both the models used in the experiment follow a multi-class one versus rest scheme with max 1000 iterations.

To see the number of false negatives predicted by our model we plot a confusion matrix for the combined dataset using Logistic Regression Model with emotionless labels as well as without them as shown in fig 2 and 3. We observe that number of true positives increase significantly when the emotionless label is removed. This implies that our performance increases in absence of emotionless labels.

6 Results

Table 2 presents the results of the Logistic Regression (LR) model on different datasets. The Titanic (T) dataset is relatively larger and less skewed than the Friends (F) and Walking Dead (WD) dataset, thus, the averaged f-score on F and WD are low. The Titanic dataset achieves good performance on almost all emotions with Fear achieving an f-score of 0.8. The frequency of anger, disgust, fear, sad and surprise emotions is quite low in Friends dataset and thus, are misclassified into either emotionless or happy. The Combined dataset (C) combines the three datasets and performs better than the F and WD datasets, but lower than the T dataset. The reason for this can be attributed to the diverse genres of the 3 videos. An interesting point to note is that happy and suprise, anger, disgust and sad form two confusion groups, but are disambiguated by our model with high accuracy.

		F-Scores	
Class	**LR**	**LSVM**	**RF**
happy	.75	.75	.72
surprise	.55	.54	.49
sad	.70	.70	.67
disgust	.54	.55	.48
anger	.62	.62	.54
fear	.77	.77	.71
emotionless	.72	.72	.69
F_{avg}	.69	.69	.64

Table 3: Comparison of Classification Models on Combined Dataset with Emotionless

Table 3 and 4 present the comparison of various classification models when emotionless class is included and when its excluded. The performance of Logistic Regression (LR) and Linear SVM (LSVM) is quite close and these two models hugely outperform the Random Forest (RF) model. For the combined dataset we get an average F-score of 69% and

		F-Scores	
Class	**LR**	**LSVM**	**RF**
happy	.78	.79	.77
surprise	.61	.60	.57
sad	.75	.74	.73
disgust	.61	.62	.60
anger	.66	.66	.62
fear	.80	.80	.76
F_{avg}	.71	.71	.67

Table 4: Comparison of Classification Models on Combined Dataset without Emotionless

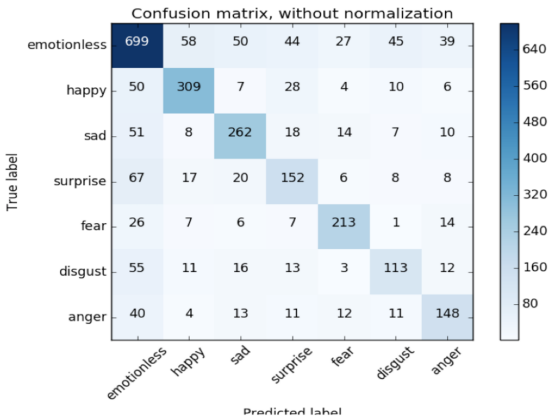

Figure 2: Confusion Matrix With Emotionless

71% with emotionless and without emotionless class respectively for both LR and LSVM whereas 64% and 67% for RF Classifier. The performance of the model improves significantly when the emotionless class is excluded from the dataset, as it is leading to high skewness.

We also tried Recurrent neural networks with LSTM units. However the results were abysmal. We obtained 25-30% accuracy with embeddings dimension of 200 when including emotionless. Due to lack of time, we did not experiment to choose the optimal set of hyperparameters like dimension of embeddings, learning rate etc. Sinha et al. (2015), used word embeddings as vectors along with a simple neural network implementation in R and obtained average 34.67% (including emotionless) and 30.04% (excluding emotionless) accuracies across the three datasets. It is clear that the amount of data we had was just not enough to learn high quality word embeddings. Thus our intuition that deep learning based approaches wont yield good results was correct.

Figure 2 presents the confusion matrix on the combined dataset with emotionless class. The results show that each emotion is well disambiguated and the classes like sad, disgust and anger are classified correctly. Figure 3 reveals that the performance significantly improves after removing emotionless class, and the average f-score increases by 2%. For the disgust emotion, many instances are classified as happy. The reason for this can be attributed to ignoring the negations in short dependencies and the low coverage of the lexicons.

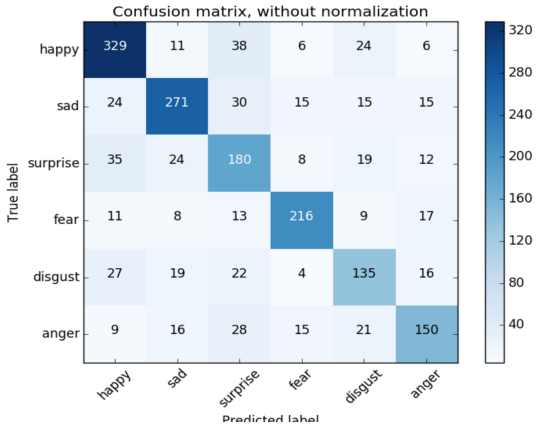

Figure 3: Confusion Matrix Without Emotionless

6.1 Feature Additive Study

In order to learn more about the impact of a feature towards the accuracy, we performed a feature additive study on all the four datasets. We used unigram+bigram as the baseline features for our model. Then we add a feature one by one in sequence of Punctuation features, POS tags, Cumulative emotion in the dialogue using EmoLex lexicon, prev_label3, prev_label2 and prev_label1 thereby learning each features contribution to the results. Figure 4 show results of this study on combined dataset with and without emotionless class using Logistic Regression model. It is observed that the prev_labels are the features which contribute immensely to results. Prev_label1 has the maximum significance followed by prev_label2 and prev_label3. This is a significant observation, as it reveals that movies follow a time-series data and thus, previous labels can be used to

predict the current label.

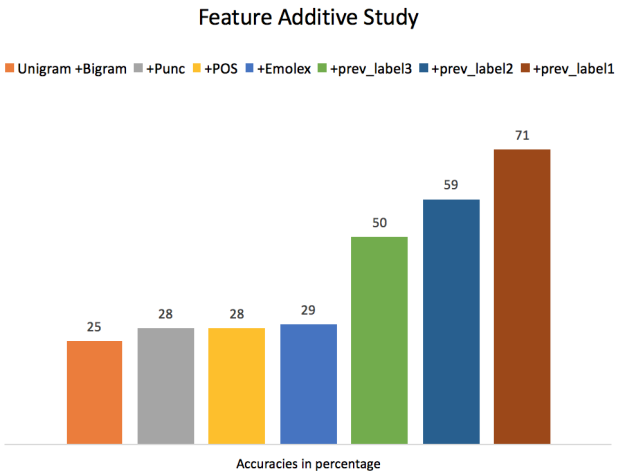

Figure 4: Feature Additive Study in Logistic Regression Model

6.2 Comparison with Previous Work

Experiments conducted by Sinha et al. (2015) to predict emotion labels for a subtitle using Random Forest and SVM had an accuracy of 33.58% and 37.65% respectively for combined dataset including emotionless class label and 25.27% and 32.9% without emotionless class label on the same dataset on which we conducted our experiments. On the contrary our experiments obtained accuracy of 64% and 69% for RF and LSVM respectively on combined dataset with emotionless label and 67% and 71% accuracy without emotionless labels. The main reason for this increase in accuracy is the variety of features we utilised in comparison to word vector embeddings.

6.3 Emotion Flow on Movie Timeline

Figure 5, 6 and 7 presents the timeline of emotions of Titanic, Friends and Walking Dead respectively. The timeline is created by grouping the scenes into k-minutes bins, where $k = 3$ for Titatic and $k = 1$ for Friends and Walking Dead, and selecting the dominating emotion for that bin. The timeline shows that the emotions are directly related to the genre of the movie, like the dominating emotions in Titanic are happy and sad, as its a romantic movie while fear is dominating in Walking Dead. The scenes before the Titanic is sinking have intense rise in fear, marked by circle on the timeline while the ending song shows the transition to happiness. The visualization of the movie on its timeline validates that its a time-series data with few emotion transitions.

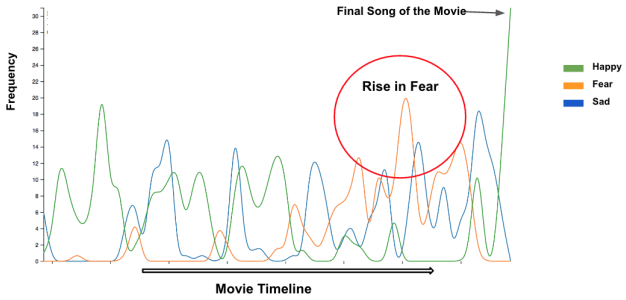

Figure 5: Flow of Emotions in Titanic

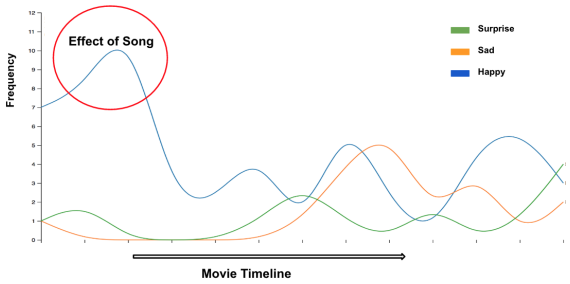

Figure 6: Flow of Emotions in Friends

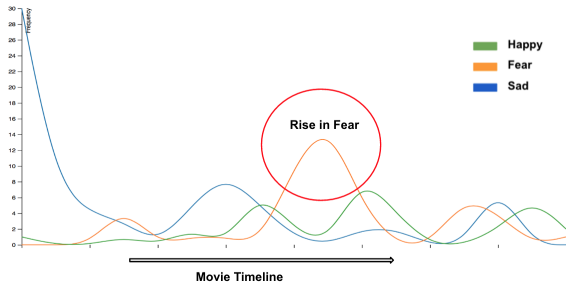

Figure 7: Flow of Emotions in Walking Dead

7 Conclusion and Future Work

In this paper, we presented an approach to tag emotion label associated with each subtitle. We contributed some interesting features like the previous k-tags. We were successful in achieving an F-score of 71.24 on the combined dataset. Because different genres of videos have different flow of emotion,

the visualizations we presented here validated the relation between genre and the emotion expressed in them.

Some of the limitations of this work include the following

- We did not use short term dependency based features that might have helped to handle negations
- Many of the subtitles in our datasets were less than 5 words long. The results we have presented here include such subtitles. We believe that the results might have improved had we set some threshold on the length of subtitles. For example, in Titanic, 20% of the subtitles were less than 5 words long
- As mentioned earlier, since our dataset was small we did not spend too much time on neural networks based models. Even though we quickly prototyped RNN-LSTM, improving the results of this model was not our top priority. One possible extension of this work includes using amazon mechanical turk to get more annotated data and then apply more sophisticated models like recursive neural networks
- From the results it is clear that POS percentages did not contribute much towards the evaluation metrics. This might be because we lemmatized POS tags. Instead we could have used for example all the different variants of NN such as NNP, NNS and similarly for other POS tags.

In the future, we plan to work on some of the above limitations

Acknowledgments

We thank Prof. Smaranda Muresan for giving us this opportunity to work on an intriguing project and for her continuous guidance and support during the development of this project over the semester.

We thank TA's Olivia and Zitong for their constant support throughout the semester.

References

Agrawal, Ameeta and Aijun. 2012. *Unsupervised Emotion Detection from Text Using Semantic and Syntactic Relations,* IEEE Publication, volume 1, Pages 346-353.

Alm, Cecilia Ovesdotter, Dan Roth, and Richard Sproat. 2005. *Emotions from text: machine learning for text-based emotion prediction,* Proceedings of the conference on human language technology and empirical methods in natural language processing. Association for Computational Linguistics.

Canales, Lea, and Patricio Martnez-Barco. 2014. *Emotion Detection from text: A Survey,* Processing in the 5th Information Systems Research Working Days.

Ekman, Paul. 1992. *An Argument for Basic Emotions,* Cognition and Emotion, volume 6, Page 169200.

Kalyan, Chetan and Min Y. Kim. 2009. *Detecting emotional scenes using Semantic Analysis on Subtitles,* Junio.

Mohammad, Saif M. 2012. *From Once Upon a Time to Happily Ever After: Tracking Emotions in Mail and Books,* Decision Support Systems, Volume 53, Pages 730-741.

Park, S.B., Yoo, E., Kim, H. and Jo, G.S. 2011. *Automatic emotion annotation of movie dialogue using WordNet,* Intelligent Information and Database Systems, volume 6592, Page 130-139.

Shelke, Nilesh. 2014. *Approaches of Emotion Detection from Text,* International Journal of Computer Science and Information Technology Research 2.2.

Sinha, Utsav, Panda, Rajat. 2015. *Detecting Emotional Scene of Videos from Subtitles*.

Strapparava, Carlo, and Rada Mihalcea. 2008. *Learning to identify emotions in text,* Proceedings of the 2008 ACM symposium on Applied computing, ACM.

Contribution : Project

- Siddharth and Tejas are worked on building the codebase to train and classify various algorithm using multiple parameters and features that are to be tested
- Keerti and Nitesh are worked on developing various combinations of features as discussed in referred papers and what we felt might boost the accuracy.
- Keerti and Tejas worked on exploring various ways to plot/ visualize emotions on timeline drawing cues from Mohammad (2012).

- Siddharth and Nitesh are working on making useful comparisons from the timeline of different movies, for e.g. comparing the emotion flow between comedy and horror movies.

Contribution : Paper

- Keerti worked on related work, visualization and putting the paper together.
- Nitesh worked on the Introduction, Lexicon, Feature Selection and Classification Experiments.
- Tejas worked on the Abstract, Classification Experiments and Results.
- Siddharth worked on the Classification Experiments, Results, Future Work
- We followed an iterative approach wherein each part of the paper was reviewed by other members which led to a discussion and revision of the content.

Appendix

CodeBase

`https://github.com/Dharamsitejas/NLP-Movie-EmotionMining`

Dataset

`https://github.com/utsavsinha/movie-emotion-mining`

Microsoft Blog

`https://www.microsoft.com/developerblog/real-life-code/2015/11/30/Emotion-Detection-and-Recognition-from-Text-using-Deep-Learning.html`