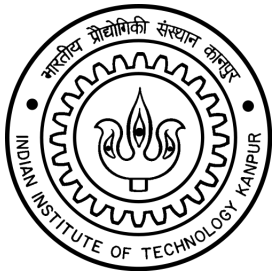
Detecting Emotional Scene of Videos from Subtitles

Utsav Sinha, 12775 Rajat Kumar Panda, 12545

17th April 2015

CS365: Artificial Intelligence
Guide: Amitabha Mukherjee



Contents

List of Figures

List of Tables

1 Motivation

With the explosion of multimedia content on the web, it becomes increasingly difficult to find the movie of one's choice. Classification of movies mainly relies on human reviews and ratings.

This project aims to automatically tag a movie and pin-point out the emotional scenes in an unsupervised manner. Finding the highlights of a video helps to get a quick review in advance or to watch the major scenes once again. This can further be extended for genre classification, recommendation systems and for detecting profanity.

2 Introduction

Videos express emotions using a variety of ways. The more prominent sources are:

- Dialogue Content
- Music
- Way of Articulation
- Facial Expressions
- Background Scene
- The Context

In this project, we are applying Natural Language Processing (NLP) techniques on video subtitle dialogues to accomplish the task of detecting emotional scenes. The dialogues of a movie form an ideal source from which the sentiments of a video can be analysed. The words being spoken along with their semantic context would help in annotating the dialogue with its appropriate emotion. The classifying emotions, referred to as major emotions includes: love, happiness, surprise, emotionless, sad, disgust, anger and fear(Table 1). These can be further classified into broader categories of positive, neutal and negative emotions.

Positive	Neutral	Negative
love (0)	emotionless (3)	sad (4)
happiness (1)		disgust (5)
surprise (2)		anger (6)
		fear (7)

Table 1: Major Emotions

[1]Numbers represents number of dialogues with a particular emotion

[2]The second row represents accuracy calculated after removing *emotionless* while first row results includes it

With extensive learning on large corpus, the pragmatic inferences along with the semantic context of phrases can be derived. This would be used to automatically annotate sentences which would then be trained on sentiment labeled dataset.

3 Dataset

Video subtitles were obtained from [pod] and [fTS]
Since subtitles with labeled emotions were not available, so a few TV-episodes and movie subtitles were hand-tagged (Table 2) with emotional labels. This relatively took a long time. The dataset formed the ground truth against which our proposed model would learn from and would later be compared with. The dataset can be found at [Sin]

Emotion	Titanic	Friends S05E14	Walking Dead S01E01
love	154	0	0
happiness	229	61	36
surprise	191	15	40
emotionless	609	126	205
sad	271	19	80
disgust	211	3	9
anger	164	11	52
fear	217	8	49

Table 2: Labeled Dataset [3]

4 Related Work

Classification of sentences into binary sentiment classes has been done extensively [WWH05] and [AXV+11], but tagging into multiple emotional clusters has not been done in detail. [PYKJ11] outlines ideas based on WordNet. [KK09] works towards the same goal but uses Naive emotion count classifier and Maximum Entropy classifier with Unigram model. Their results are in Table 3.

The results included here grouped 15-20 sentences into a scene before calculating accuracy, whereas our approach calculated accuracy based on individual sentences, thereby giving poorer results (Table 4). Moreover, we used eight major emotion clusters instead of six.

We did not use Maximum Entropy Model as learning similarity between words is done by Word2Vec whose skip-n gram model captures context more

[3] Numbers represents number of dialogues with a particular emotion

Movie	# of Sentences	# of Groups	Accuracy
Gladiator cd1	1260	62	72%
Love Actually	1829	91	48%
Troy	1156	57	65%
X-men 1	689	35	62%

Table 3: Accuracy with Movie Subtitles

accurately. Naive count classifier does not uses any learning from the labeled data. We have also used an equivalent non learning algorithm by using raw word2vec vectors.

5 Approach

Implementation Overview

A bag of words model was used where words were converted into vector representation. These word vectors helped to construct the sentence vector of a dialogue. Pre-processing was done on word vectors and subtitle files for refining the sentence vector. The sentence vector was then sent to learning phase where learning algorithms were used to classify a dialogue into one of 8 emotion categories. Figure 1 presents the overall overview of the implementation.

5.1 Vector Representation of Words

To get a vector representation of words, Wikipedia corpus was used in training by Word2vec [MCCD13]. Word2Vec is a skip-gram neural network implementation which uses context of sentences to build vectors in an unsupervised manner. This provided a 100 dimensional vector for each word(with at least 5 occurrences) in the training corpus.

An n-gram uses a continous sequence of n words for model creation. Skip-gram model is a generalization of n-gram model where the words do not need to be consecutive in the sentence for constructing the model. So words can be skipped over and resulting vectors can be used for word arithmetic [MSC+13]. This is particularly useful in sentiment analysis where separators like conjunctions, prepositions can be skipped to bring together the sentiment words. For example:

I am overwhelming *glad* to see you *happy*.

A skip gram model will result in bringing together *glad* and *happy* in their vector representation.

The labeled data is now used for sentiment tagging (Figure 1). The vectors of each word in a dialogue are summed and its average is send as the sentence vector. along with the labeled emotion to the learning phase.

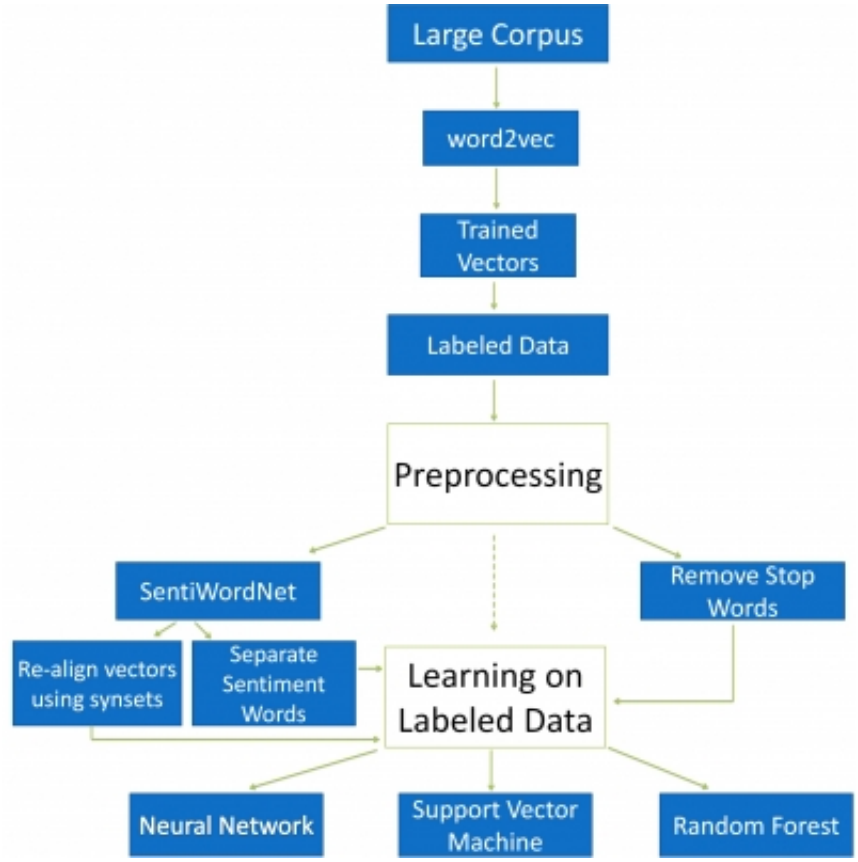


Figure 1: Implementation Overview

5.2 Pre-processing Stage

The word vectors obtained above and labeled data is pre-processed before it is sent to learning phase. For that, we used three heuristics which can be used independently of each other.

5.2.1 Reweighting Word Vectors

Reweighing of the vectors of major emotions was done as a testing parameter. This step is essential since emotion such as *happiness* is not expressed by the literal use of word happiness alone but also by semantically close words like

pleasant, *delightful*, *cheerful* among many other. So these synonyms should also be able to influence the vector of the major emotions.

Taking inspiration from [PYKJ11] where similar emotions were grouped together with the help of a dependency graph built from SentiWordNet, we adopted a similar approach.

SentiWordNet [BES10] provides synonym sets (*synsets*) of words and a sentiment score. Sentiment score includes positive score (pos), negative score (neg) and an objective (neutal) score (obj) such that:

$$pos + obj + neg = 1$$

Taking n as the number of all synonyms taken from all the synsets of a major emotion *emo*, vector V_{emo} is recalculated using:

$$V_{emo} = \frac{V_{emo} + \sum_{i=1}^{n} \alpha_i V_i}{1 + \sum_{i=1}^{n} \alpha_i}. \tag{1}$$

$$\alpha_i = max(pos_i) \tag{2}$$

$$\alpha_i = max(neg_i) \tag{3}$$

Here, α_i is the peak sentiment score of a synonym word i. For negative major emotion like anger, fear, sad and disgust Equation 3 is used while for positive major emotion like happines, love and surprise, Equation 2 is used while calculation Equation 1.

5.2.2 Emphasizing Words in Labeled Data

Stop words such as *he*, *there*, *because* etc were removed from the labeled data since they do not contribute to the overall sentiment of a sentence. More emphasis was given to synonyms of sentiment words taken from WordNet[4] [Mil95] .

NLTK library [BKL09] was used with Python for implementing these.

5.2.3 Calculating Sentence Vector

After irrelavant words were removed, the vector representation of the sentence was constructed from the word vectors of the remaining words. For this, simple averaging of the word vectors was used (Equation 4).

$$V_{sen} = \frac{\sum_{i=1}^{n} V_i}{1 + \sum_{i=1}^{n} i}. \tag{4}$$

This sentence vector along with the labeled emotion(hand labeled ground truth) was sent to the learning algorithm.

[4]Since SentiWordNet contains fewer words than WordNet, so sentiment words taken from SentiWordNet were also searched for their synonyms in WordNet to emphasize their word vectors

5.3 Learning Algorithms

Learning on labeled data was done using:

Support Vector Machine (SVM) Classification
Multiclass SVM was used with Linear, Radial Basis and Polynomial Kernel. The implementation was in **Python**.

Random Forest
Random Forest based implementation of decision trees was used. The implementation was in **R** language.

Neural Networks
Simple neural network based implementation in **R** was used.

A no learning algorithm using raw Word2Vec vectors was also used where cosine similarity of sentence vector with the major emotion vectors was minimized to find the emotion of a dialogue.

6 Result

SVM with Linear, Polynomial and Radial Basis Kernel were compared with Random Forest, Raw Word2Vec vectors and Neural Network Implementations. 10-fold cross validation was used to calculate accuracy of models. Table 4 shows the accuracy obtained by different algorithms. The accuracy after removing the emotionless group is also stated in the Table 4.

Video	Raw Word2Vec	Neural Network	SVM	Random Forest
Titanic	32.34	38.21	39.74	36.46
	20.32	27.80	34.70	24.80
Walking Dead	33.20	36.63	38.93	34.72
	24.78	29.14	31.52	26.41
Friends S05E14	23.31	27.56	32.10	29.76
	17.78	21.33	25.31	22.92
Overall	29.82	34.67	37.65	33.58
	21.44	30.04	32.90	25.27

Table 4: Accuracy of Emotion Detection in Percentage [5]

[5]The second row represents accuracy calculated after removing *emotionless* while first row results includes it

Some dialogues succesfully classified by our project and their ground truth along with other cases which were mis-classified are in Table 5 and Table 6.

Titanic:	Do me the honor. And never let go of that promise. I'll never let go, Jack. I'll never let go.	Love
Titanic:	Knock it off. You're scaring me.	Fear
Titanic:	Seeing her coming out of the darkness like a ghost ship ...	Sad
Titanic:	Like I told you, I go to America to be millionaire.	Happiness
Titanic:	My heart was pounding the whole time. It was the most erotic moment of my life ...	Love
Titanic:	You can't keep us locked in here like animals. The ship's bloody sinking.	Fear
Titanic:	And all the while, I feel I'm standing in the middle of a crowded room ... screaming at the top of my lungs, and no one even looks up.	Disgust
Walking Dead:	We didn't have a great night. - Look, man, I may have a...	Sad
Walking Dead:	You pull the trigger, you have to mean it. Always remember that, Dwayne.	Fear
Friends:	All right, if he wants a date, he's going to get a date.	Happiness

Table 5: Dialogues classified succesfully

Video	Dialogue	Gold Truth	Predicted
Titanic:	Get the master-at-arms. Now, you moron!	Anger	Emotion-less
Titanic:	You're gonna die an old lady, warm in her bed.	Sad	Fear
Titanic:	Wait, I don't have to leave. This is my part of the ship. You leave.	Anger	Surprise
Titanic:	But my mother looked at him like an insect. A dangerous insect which must be squashed quickly.	Disgust	Fear
Titanic:	You're so stupid, Rose. Why did you do that? Why?	Surprise	Sad

Table 6: Dialogues which were not classified well

6.1 Reasons for mis-classification

We used a bag of word representation where aggregate of words represent the emotion of the dialogue. So a sentence like *You're gonna die an old lady, warm in her bed.* which represents a *sad* scene in the movie gets classified as *fear* due to prominance of word *die* and *old*.

Emotions like anger depends more on the way of articulation and stress given to the words rather than the choice of words. This **prosody** can be captured more accurately in *speech analysis* than text processing.

A break-up of Accuracy vs Emotion is in Figure 2. This highlights that emotions like *happiness* and *sadness* are classfied more accurately than emotions like *anger* and *disgust* which involves a lot of irony, sarcasm and prosody. These can be easily understood by humans but it is difficult for a machine to decipher its true meanings.

Lower performance on comic video *Friends* is due to dominance of negative emotions in training data (See Table 2) and selection of major emotions (4 of the 8 are negative). Moreover, friends involves lots of indirection and sarcasm which is difficult to detect.

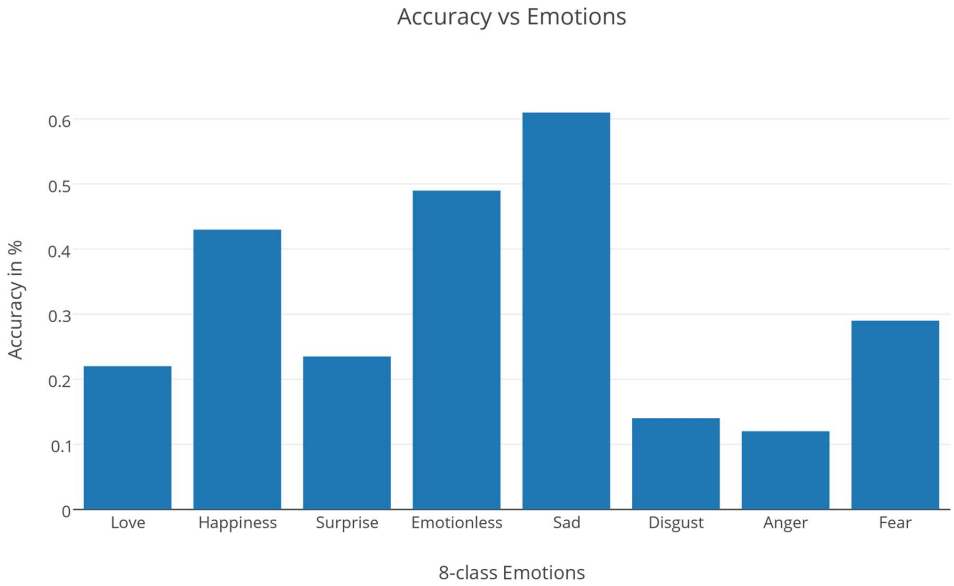


Figure 2: Accuracy vs Emotion

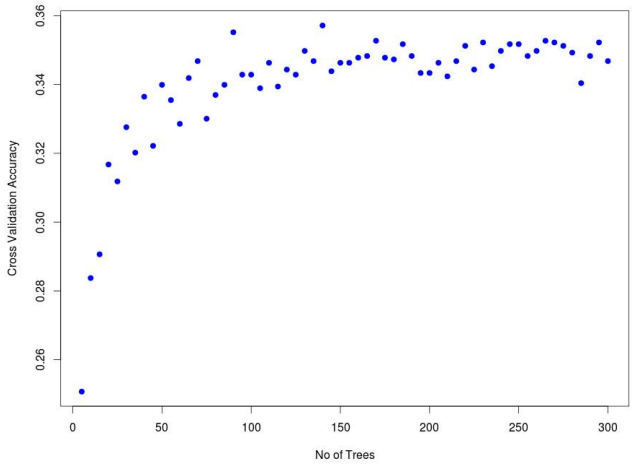


Figure 3: Finding the Elbow in Random Forest

Figure 3 finds out the elbow for the number of trees in random forest. 140 turned out to be the optimum number of trees beyond which accuracy becomes asymptotically constant.

The confusion matrix of Linear SVM (Figure 4) and Random Forests (Figure5) shows that Linear Kernel SVM outperforms the other methods. See Table 2 for emotion legends.

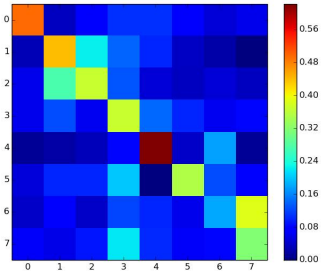


Figure 4: Confusion Matrix for SVM

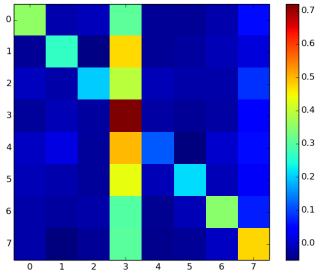


Figure 5: Confusion Matrix for Random Forest

7 Future Improvements

More labeled data could be used for training.
New metric can be used for calculating sentence vector.[LM14]

References

[AXV+11] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[BES10] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, 2010.

[BKL09] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

[fTS] Tvsubtitles.net Subtitles for TV Shows. Subtitles for tv shows - http://www.tvsubtitles.net.

[KK09] Chetan Kalyan and Min Y Kim. Detecting emotional scenes using semantic analysis on subtitles, 2009.

[LM14] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Mil95] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[pod] podnapisi. Subtitles datasource - http://www.podnapisi.net.

[PYKJ11] Seung-Bo Park, Eunsoon Yoo, Hyunsik Kim, and Geun-Sik Jo. Automatic emotion annotation of movie dialogue using wordnet. In *Intelligent Information and Database Systems*, pages 130–139. Springer, 2011.

[Sin] Utsav Sinha. Subtitle files labeled into 8 emotional groups - Labeled Data.

[WWH05] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.