Sentiment Classification with Thwarted Text

Thomas Fitzgerald
New College of Florida

Mei Maddox
New College of Florida

ABSTRACT

We classified documents based on sentiment by breaking each document into clauses, then scoring each word by VAD criteria, with a goal of correctly classifying documents containing sentiment thwarting. We attempted two models, a LSTM neural network and a simplistic rules based approach.

1 INTRODUCTION

Thwarted Expectations, or Thwarting, is a type of issue in sentiment analysis where a document has multiple lower-level elements of one polarity but has the opposite polarity at a higher level[1]. It should be noted that it is distinct from sarcasm, which is a text of one apparent polarity, but understood to be of the other polarity based on context. Thwarting typically occurs due to an explicit opposite polarity statement about a 'critical element' overcoming the more frequent polarity applied to less-important elements[1]. However, defining what makes one element 'critical', as opposed to other elements, generally requires domain knowledge in the form of some type of hierarchy. We attempt to find a method that is not based on explicit domain hierarchies.

One of the difficulties regarding thwarting is that it is rare, making up approximately 1-2% of most datasets, if that.[1] Identifying thwarting via hand annotation is work-intensive, and the lack of prepared datasets make researching it difficult. Based on a review of available examples, we hypothesize that thwarting usually occurs at the tail end of a document, which forms the basis for our rule-based model.

During this process, we utilized Valence-Arousal-Dominance (VAD) criteria for word-level scoring, using the NCR-VAD lexicon[2]. Valence, or overall pleasantness, was the main factor in our scoring, with arousal and dominance acting as modifiers.

2 RELATED WORK

Detecting Turnarounds in Sentiment Analysis: Thwarting describes implementation of a rule-based system, utilizing domain knowledge on Amazon camera reviews. The study used a small dataset (21 reviews), identified by a group of 3 annotators[1].

Thwarting is a rare occurrence in most corpi, and research on the topic is limited. Despite its low apparent prevalence, this type of thwarting could contribute to the false positive rate in sentiment classification, and methods for identifying it could lead to better understanding of user sentiment.

3 DATASET

Sentiment Analysis of IMDB Movie Reviews is a dataset of 50k English language movie reviews from imdb.com[3]. Each review is a string, with either a 'positive' or 'negative label. The dataset contains exactly 25,000 positive reviews, and 25,000 negative reviews, based on this labeling. Reviews cover multiple types of media, including series and video games, but movie reviews are the most common.

Although the dataset comes pre-labeled for general sentiment, for our experiment, we needed labeled examples of thwarting. We read reviews and then hand annotated each as 'thwarted' or 'normal'. Additionally, we marked whether the review read as having mixed or unmixed sentiment, based on annotator opinion. After reading approximately 700 reviews, we had found 10 examples of thwarting, 4 negative and 6 positive. We then randomly selected 20 mixed sentiment reviews, and 20 unmixed sentiment reviews from the 700 we had read. This gave us a dataset of 50 labeled reviews.

4 METHODOLOGY

4.1 Cleaning and Pre-Processing Pipeline

After our labeled dataset was generated, we saved each review into a separate text file, with index and labeling information stored in the file name.

We converted all text to lowercase, removed new-line and new-paragraph characters, and separated all number and alphabetical characters to avoid later issues with tokenization.

Using the spacy library[4], we applied part-of-speech tagging, and identified sentences based on standard sentence-ending punctuation before subdividing each sentence into clauses.

A clause was defined as a sub-section of a sentence containing a conjunction ('CC' or 'IN') that either begins the sentence (e.g. "Although ...") or is surrounded by 2 sets of tokens with disjointed POS tags. This method avoids defining lists as separate clauses. For example, "...are terrible and the directing…" would be split into two clauses but "bad actors but good story" would remain one.

Each list of clauses was then lemmatized, using the POS tagging from the previous step to improve accuracy.

A VAD score was then mapped to each word. For words which do not exist within the VAD lexicon, an approximate score was calculated based on semantically-similar words. Word similarity was gauged using cosine similarity of GLoVe vector representations. If no, appropriate VAD vectors were found within the k=10 most similar words, then the word was simply mapped to the VAD averages across the entire clause. The averaging technique assumes only one sentiment is expressed per clause.

A method was then applied for dealing with negation, based on a modified approach from Pang et al(2002). In our application, all tokens between the negation "not" word and the following non-apostrophe punctuation or clause end, had their valence inverted, so .9 (highly pleasant) would become 1-.9=.1 (highly unpleasant). Furthermore, the token immediately following "no," "neither," and "nor" had their valence inverted.

We then removed all stopwords and pseudowords. Any remaining words with a a valence between .3 and .7 (neutral), or arousal/dominance below .1 were removed. Each clause was then defined as the mean average of valence*arousal for all words. This single score was then the value for that clause, and all the clauses for each document were stored as a vector of those values.

4.2 Lexicons

We used two lexicons for this project: the NRC-VAD Lexicon and the GloVe lexicon[2][5]. The GloVe lexicon was modified into a BallTree and converted to a .pkl file.

4.3 LSTM

Long Short Term Memory recurrent neural network models are a particular type of model which can handle long sequences of data. We theorized that the LSTM could extract underlying patterns of sentiment progressions within a review to more accurately predict overall sentiment, regardless of thwarting

We used a fairly standard torch LSTM with 5 layers and a hidden size of 10. We believe this method may be effective on a larger data set, but were unsure about the results on the small dataset we had available.

4.4 Rule-Based Model

This method was based on the hypothesis that most thwarting occurs at the end of a document, and has the following fundamental steps.

(i). Divide the clauses within each document into two ordered (non-shuffled) sections at ratio of 80:20. However, if the total number of clauses within the document is less than 5, simply average the entire document.

(ii). Calculate the average sentiment for each of the two sections individually. If the second section contained more than three clauses, calculate a weighted average with additional priority the final-clause sentiment by loading the section with n/2 additional copies of the final vector score. So, for example:

- second vector = [.8,.2,.4,.4,.9]
- second vector = [.8,.2,.4,.4,.9] & [.9]*(2//5)
- average = mean([.8,.2,.4,.4,.9,.9,.9])

The goal of this step is to increase the weight of the final clause proportional to the length of the document.

(iii). Translate the average sentiments into binary representations based on a 0.5 threshold value

(iv). If the two sections have different binary sentiments *and* the difference between their averages is greater than

0.2, thwarting is assumed and the second section score is returned. Otherwise, the first section score is returned.

5 RESULTS

5.1 Rules-Based Model

PREDICTION		Positive	Negative
TRUTH	Positive	23	1
	Negative	21	5

Precision: 0.52 Recall: 0.96, Accuracy: 0.56

The above table shows the performance of our rules-based model at overall sentiment prediction across the entire dataset.

PREDICTION		Positive	Negative
TRUTH	Positive	6	0
	Negative	2	2

Precision: 1.0 Recall: 0.75, Accuracy: 0.8

The above table is overall sentiment prediction performance for only the 10 thwarted documents in our dataset.

PREDICTION		Thwarted	Normal
TRUTH	Thwarted	1	9
	Normal	2	38

Precision: 0.1 Recall: 0.33, Accuracy: 0.78

The above table is our model’s performance at predicting whether or not a document is thwarted. The dataset is 80% ‘normal’, so 78% accuracy approximates a random guess.

5.2 LSTM

The LSTM model classified all sentiments as positive.

6 DISCUSSION / FUTURE WORK

From an accuracy perspective, our rules-based model approximated a random guess. However, certain settings (a lower valence difference threshold primarily) improved it's performance as a thwarting classifier somewhat, at one point managing a .44 precision rate. Our dataset is really too small to verify if this performance is generalizable, but it may be worth pursuing as an initial "filtering" step before hand annotation. Since thwarting occurs at a rate of ~1-2\% in most datasets, extracting a subset of even 20-30\% thwarted documents could be useful.

From a dataset perspective, many of the reviews we were dealing with were length, a problem compounded by the clause-level tokenization we used. Although clause-level separations contain useful information, applying some kind of intermediate averaging (for example, subdividing the document into 4 sections, and averaging each section), then running our models, may have been more effective.)

Our LSTM model was severely dysfunctional. This may have been a result of the extremely small dataset or simply a poor method of representing sentiment progression within a single document.

The most important thing going forward is to get a larger labeled dataset of thwarted documents. Ten is not nearly enough to learn much from.

7 APPENDIX

7.1 VAD Scoring

From an accuracy perspective, our rules-based model approximated a random guess. However, certain settings (a lower valence difference threshold primarily) improved it's performance as a thwarting classifier somewhat, at one point managing a .44 precision rate. Our dataset is really too small to verify if this performance is generalizable, but it may be worth pursuing as an initial "filtering" step before hand annotation. Since thwarting occurs at a rate of ~1-2\% in most datasets, extracting a subset of even 20-30\% thwarted documents could be useful.

From a dataset perspective, many of the reviews we were dealing with were length, a problem compounded by the clause-level tokenization we used. Although clause-level separations contain useful information, applying some kind of intermediate averaging (for example, subdividing the document into 4 sections, and averaging each section), then running our models, may have been more effective.)

Our LSTM model was severely dysfunctional. This may have been a result of the extremely small dataset or simply a poor method of representing sentiment progression within a single document.

The most important thing going forward is to get a larger labelled dataset of thwarted documents. Ten is not nearly enough to learn much from.

7.2 Clauses

One major hurdle we encountered in this project was determining how to separate each review. In the case of thwarting, granular sentiment plays an extremely important role in identification for a model. A common general format for a thwarted text is multiple clauses in one polarity, followed by a clause of the opposite polarity on an important topic. If a text has four sentences, three focusing on individual elements of a movie negatively (bad acting, bad directing, bad effects), followed by a summary sentence (“I loved it!”), the total amount of negative words in the document is likely to be higher than the positive word count. In a simple bag of words model, this gives us an overall negative sentiment.

Because of this, it’s important to process each clause as an ordered set, and determine the sentiment of each clause as a whole (Negative,Negative,Negative,Positive). Now, we can reference the order and type of the sentences: most thwarting involves a chain of clauses in one polarity, followed by a sentence of the opposite.

However, sentiment can be identified at a more granular level than by sentence. A complex sentence, broken up into separate clauses, can express both positive and negative views on different entities within the same sentence. This led us to try breaking up each sentence into a set of clauses. We initially tried a rule-based approach, using punctuation (,;:) and keywords (but, however, etc.), but had poor results, partially due to the simplistic method, and partly due to poor grammar in the training data.

We then decided to try using human annotators to separate and classify each clause. First, the document was fed through an extremely simplistic sentence parser to break the text into multiple lines. Next, the human annotator would further break each line into clauses, while correcting any errors made by the sentence parser. Last, the annotator would mark each clause as positive or negative, and store the results as a text string in a separate file. Both group members and a conscripted third person acted as annotators. Each person was assigned 15 of 50 documents, with 5 additional documents assigned to all 3 annotators.

We started with the joint documents, and had a fourth person compare each version for similarity in clause splitting and sentiment rating. The sentiment ratings were very similar (lowest Cohen's Kappa coefficient of 0.8649 between pairwise combinations of annotators), but there was strong disagreement on how to separate clauses. In one example, containing 8 sentences, one annotator marked 10 clauses, one 11, and one 18. However, nearly all the corresponding text was sentimentally identically even if the text was attributed to different clauses.

Based on this, we determined that human annotators were reasonably consistent at rating the polarity of a segment of text, but extremely inconsistent at defining a clause. The overall arrangement of labels for each text remained consistent: multiple positive clauses, followed by a mix of positive/negative (mostly negative), and then finishing with a positive. We decided to drop the hand-annotation option for clauses, and went with a rule-based approach we could incorporate into our pipeline.

8 REFERENCES

[1] Ramteke Ankit, Malu Akshat, Bhattacharyya Pushpak; Nath, Saketha. 2013. *Detecting Turnarounds in Sentiment Analysis: Thwarting* https://aclanthology.org/P13-2149.pdf

[2] Mohammad Saif. 2011.NCR Valence, Arousal, and Dominance (NCR-VAD) Lexicon. https://saifmohammad.com/WebPages/nrc-vad.html

[3] LAKSHMIPATHI N. 2020. *Sentiment Analysis of IMDB Movie Reviews*. https://www.kaggle.com/code/lakshmi25npathi/sentiment-analysis-of-imdb-movie-reviews

[4] Spacy en_core_web_sm pipeline, https://spacy.io/models/en

[5] Pennington Jeffrey, Socher Richard,Manning Christopher. 2014. *GloVe: Global Vectors for Word Representation*. https:/nlp.stanford.edu/projects/glove/

[6] Pang Bo,Lee Lillan,Vaithyanathan Shivakumar. 2002. *Thumbs Up? Sentiment Classification using Machine Learning Techniques.*

https://ncf.instructure.com/courses/6788/assignments/53643?module_item_id=154725

[7] Wankhade Mayur,Rao Annavarapu Chandra Sekhara,Kulkarni Chaitanya. 2022. *A survey on sentiment analysis methods, applications, and challenges* https://link.springer.com/article/10.1007/s10462-022-10144-1

[8] Panda Saismita, Gupta Saumya, Kumari Swati, Yadav Parul. 2020. *Sentiment Analysis Techniques and Approaches* https://www.ijert.org/research/sentiment-analysis-techniques-and-approaches-IJERTV9IS060350.pdf