UNIVERSITY OF SOUTHAMPTON

Prediction of Reader Emotion in Stories

by

Ashwathy.T.Revi

Supervisor: Dr. David E Millard

Examiner: Dr. Richard A Watson

A thesis submitted in partial fulfillment for the
degree of MSc in Artificial Intelligence

in the
Faculty of Physical Sciences and Engineering
School of Electronics and Computer Science

November 2019

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

MSc in Artificial Intelligence

by Ashwathy.T.Revi

Stories differ from other types of text in many aspects like the presence of a protagonist and varying empathy of the reader towards different characters. General emotion analysis techniques do not take this into account. Incorporating these factors into emotion detection might improve reader emotion prediction. A series of out of the box NLP techniques were used to extract information about characters and events from the stories. Using this information, a novel approach to predict the emotion felt by the reader on encountering a particular event in a story is presented. This is a heuristic that can be part of a rule based classification system. Its performance compared against a general rule based emotion analysis system. The results show that though the rule succeeds in correctly classifying a few instances where the general purpose algorithm fails, it also makes many mistakes. Mistakes arise due to errors in the intermediate NLP steps, causing a drop in accuracy measures. An analysis of the mistakes suggests that errors in the Sense Disambiguation step and in the +/- effects lexicon have more effect than the ones in other parts of the pipeline. When the +/- lexicon is substituted to use a manually annotated gold lexicon, an improvement is observed, suggesting that the method can be useful if accuracy of intermediate steps can be improved but not otherwise.

Contents

List of Figures

List of Tables

Acknowledgements

Thanks to Dr David Millard for his supervision and support. Thanks to University of Southampton for the opportunity to do this project.

Statement of Originality

- I have read and understood the ECS Academic Integrity information and the University's Academic Integrity Guidance for Students.
- I am aware that failure to act in accordance with the Regulations Governing Academic Integrity may lead to the imposition of penalties which, for the most serious cases, may include termination of programme.
- I consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to verify whether my work contains plagiarised material, and for quality assurance purposes.

You must change the statements in the boxes if you do not agree with them.

We expect you to acknowledge all sources of information (e.g. ideas, algorithms, data) using citations. You must also put quotation marks around any sections of text that you have copied without paraphrasing. If any figures or tables have been taken or modified from another source, you must explain this in the caption and cite the original source.

I have acknowledged all sources, and identified any content taken from elsewhere.

If you have used any code (e.g. open-source code), reference designs, or similar resources that have been produced by anyone else, you must list them in the box below. In the report, you must explain what was used and how it relates to the work you have done.

I have not used any resources produced by anyone else.

You can consult with module teaching staff/demonstrators, but you should not show anyone else your work (this includes uploading your work to publicly-accessible repositories e.g. Github, unless expressly permitted by the module leader), or help them to do theirs.For individual assignments, we expect you to work on your own. For group assignments, we expect that you work only with your allocated group. You must get permission in writing from the module teaching staff before you seek outside assistance, e.g. a proofreading service, and declare it here.

I did all the work myself, or with my allocated group, and have not helped anyone else.

We expect that you have not fabricated, modified or distorted any data, evidence, references, experimental results, or other material used or presented in the report. You must clearly describe your experiments and how the results were obtained, and include all data, source code and/or designs (either in the report, or submitted as a separate file) so that your results could be reproduced.

The material in the report is genuine, and I have included all my data/code/designs.

We expect that you have not previously submitted any part of this work for another assessment. You must get permission in writing from the module teaching staff before re-using any of your previously submitted work for this assessment.

I have not submitted any part of this work for another assessment.

If your work involved research/studies (including surveys) on human participants, their cells or data, or on animals, you must have been granted ethical approval before the work was carried out, and any experiments must have followed these requirements. You must give details of this in the report, and list the ethical approval reference number(s) in the box below.

My work did not involve human participants, their cells or data, or animals.

Chapter 1

Introduction

Stories are often designed to follow certain dramatic and emotional arcs. This means that author tries to introduce a pattern in the rise and fall of level of drama and emotion to make stories more interesting. A classic example is Freytag's pyramid Freytag (1900). First there is an exposition phase with low drama, then the action rises, reaches the climax and then falls and settles again in the denouement phase.

Interactive fiction works and games usually involve non linear narratives. The story experienced by a player depends on his choices in the game. Authoring such non linear stories that are interesting along all possible story paths highly challenging since this requires the author to simultaneously visualize the dramatic arc along all story paths. It becomes increasingly hard as complexity of the story increases. Some existing works like Mateas and Stern (2003) use systems called Drama Managers to take care of this. They require the author to tag events with their "interestingness" or "importance".

Estimating the interestingness of an event itself can be challenging for the author in non linear stories. This is because interestingness of an event can depend on many contextual elements and context can vary along different story paths. For example, the death of a character A could be considered interesting. But in a particular branch of the interactive story, that character may be a minor one, having appeared only a few times. In such branches, this event is not as dramatic as say, in a branch where A is the hero or the villain. Artificial Intelligence(AI) can potentially be used to infer interestingness of an event along all the story paths it is on and assist the author in situations like this, by computing drama and emotion arcs and providing feedback to the author.

"What is dramatic?" as outlined in Freytag (1900) is hard to define computationally. In it, he says,

> *".. a passion which leads to action is the business of dramatic art; not the presentation of an event itself, but for its effect on a human soul is the dramatist's mission."*

"Passion leading to action" loosely translates to character goal modeling and the closest we can come to modelling "effect on a human soul" is by modeling the emotion of the reader when reading the story. Other works like Abbott (2008) identify other components that give rise to dramatic interest like surprise, suspense, curiosity, action, tension, humour and so on. While there are some attempts to generate dramatic and suspenseful stories like Doust (2015), estimating these components from plain text is highly challenging. Emotion analysis, on the other hand, is highly researched and might be the easiest to determine computationally. Modeling reader emotion arcs along different story paths can also be useful feedback for the author as they may expect it to rise and fall similar to dramatic effect and alternate between positive and negative emotions.This project focuses on the reader emotion detection.

Most attempts at sentiment and emotion analysis in text use general lexicon based and machine learning techniques. It is worth noting here, that sentiment analysis, though similar, differs from emotion analysis. Sentiment or opinion detection tries to predict the general sentiment expressed in the text by the reader. Emotion analysis tries to detect the emotion expressed. Reader emotion and writer emotion also needs to be distinguished. The problem addressed in this project differs from general emotion analysis works in that it tries to predict the emotion felt by the reader and not the emotion expressed in the text by the writer. While these two may be closely related for texts like reviews or comments, they differ significantly in case of stories, depending on the context.

For example, a sentence "A kills B" might express negative emotion but if B is the villain, the reader experiences a positive emotion on reading it. This becomes complicated because, if A is the hero and B is the villain it induces a positive emotion in the reader but if B is the hero and A is the villain, a negative emotion is induced. Either way, the event A kills B will be classified negative by general emotion / sentiment analysis techniques due to the negative nature of the word "kill". The context matters here, but it is not taken into consideration. Machine Learning or deep learning techniques might be able to pick on these things if trained well, but the lack of large story datasets annotated with reader emotion makes this difficult. Most work on emotion analysis has been carried out on reviews, blogs, tweets and news stories. Comparatively, there is very little on literary texts. While news stories have some characteristics of fictional stories – like existence of characters and chronological events, their nature still differs. However, unlike literary texts, in new stories, there are no character archetypes, there is no dramatic structure of events, the number and types of events is limited and the way in which emotions are induced in the reader is very different.

Even though there is very little done on emotion analysis in stories, there is a lot of research on information mining from stories, like extracting sentiment networks, plot structure and identifying character archetypes. Such methods could potentially be incorporated into the emotion prediction process to improve it, but not much work has been done in this direction.

This project investigates ways to predict reader emotion given these challenges and opportunities. It proposes a rule that could be part of system of rules in a rule based classifier, implemented using Natural Language Processing (NLP) techniques. The research question addressed is if using this particular heuristic rule can improve reader emotion prediction in stories when used in combination with a general rule based classfier like Vader presented in Hutto and E.E. (2014).

Remaining sections are oraginized as follows: Chapter 2 provides an overview of literature from various related fields that could potentially be brought together and used. Chapter 3 describes the methodology adapted for design , implementation and evaluation explaining what choices were made and why. The emotion detection algorithm used is explained and its working is demonstrated with a small example. Chapter 4 reports the results observed and presents the analysis. Chapter 5 concludes, summarizing the work, discussing its outcome ,significance, limitations and future work.

Chapter 2

Literature Review

2.1 Narrative Analytics

The idea of analysing different story paths and collecting feedback to assist the author was proposed in Millard (2018) . It proposes using machine readerbots to simulate readers and extract all story paths from an interactive fiction game. If integrated into an authoring tool, several issues can then be identified and provided as feedback to the author. It talks about identifying structural issues like dead ends, unreachable nodes and unbreakable cycles and issues with use of locations in StoryPlaces from Millard (2017). It also covers some issues with drama and experience like if the length of each path is reasonable and if all points of interest are covered. It does not talk about drama analysis or emotion analysis in the context mentioned earlier. AI can be used to extract more sophisticated metrics like this. Emotions induced in the reader along different story paths can be predicted this way. Using Readerbots, all story paths can be obtained and emotional arcs along all paths can be drawn. Sanghrajka and Kapadia. (2017) presents Lisa which is a similar tool. It is a story assistant that assists authors by using AI to make inferences and point out inconsistencies in narratives. It also builds a knowledge base that can be queried in natural language. Marti and Gross. (2018) is a work with movie scripts that allows to make visualizations of the plot and timeline.

2.2 Emotion Analysis

Sailunaz and Alhajj. (2018) is a recent survey of existing research in emotion analysis from text and speech. It lists the different categorical and dimensional models that are used to represent emotion. Categorical models are those that have discrete categories or classes corresponding to a finite set of emotions. Ekman (1992) concludes that there are 6 basic emotion classes, anger, disgust, fear, happiness, sadness, and surprise. This

model has been used in many emotion classification works like Agrawal (2012) and Li (2014). These types of models have limits in that intensity or combination of emotions cannot be represented. Dimensional models on the other hand allow an emotion score along different dimensions. For example, The Valence-Arousal-Dominance model proposed in Mehrabian (1980) has 3 dimensions. Valence represents polarity (positive or negative) of the emotion, Arousal indicates level of excitement and Dominance indicates level of control of the emotion. Ortony (1988) model is a different approach used to generate emotion in agents. It has about 22 emotion categories whose strength is modelled as depending upon events, agents and objects in the environment of the agent exhibiting the emotion. This survey also has a list of notable papers published in this area presented as table showing features and limitations of each system and methods and emotions models used in each system. It classifies methods as keyword-based, lexicon based, machine learning based or hybrids. In Keyword based methods, there is a list of keywords associated with each emotion. lexicon-based methods are similar except that instead of keywords, a lexicon annotated with emotions is used. Several supervised and unsupervised models are also listed. It notes that unsupervised methods are more general but supervised methods achieve higher accuracy. Hybrids methods use a combination of all this. The accuracy of these different methods are compared in previous surveys Yadollahi (2017), Binali H (2010), Canales L (2014), CR (2015) and Tripathi V where they found lexicon based methods achieved higher accuracy. Several rule-based and semantic techniques are also used in some works. For example in Grover S (2016) NLP is used to extract features which are then passed through a rule engine to filter out emotion classes. Vader from Hutto and E.E. (2014) is another rule based model and is implemented in python's NLTK package presented in Bird and Klein. (2009). In Hutto and E.E. (2014) comparison of performance on a dataset of tweets with several other state of the art benchmarks is presented where they show that Vader performs better.

2.3 Reader emotion prediction

Compared to the amount of literature on emotion analysis, there is very little that addresses reader emotion as opposed to writer emotion. Ye and Xu (2012) compares many techniques to predict reader emotion for new articles and find that random k-label sets classifier works best, when used with features derived from intersection of chi-square statistics and document frequency. In their later work, Ye and Xu (2013), they develop a model where features are extracted using weighted LDA and use a multi-label k-nearest neighbor model (MLkNN) to predict reader emotion and observe better results. Lin and Chen. (2008) and Bhowmick and Mitra. (2010) also studies reader emotion but for news articles as well. Yang and Chen. (2009) checks for relationship between writer and reader emotion and find that in news articles positive writer emotion tends to illicit

positive reader emotion, but in blog posts, it depends on the topic. Dramatic stories or literary works are not investigated here either.

2.4 Emotion Analysis in stories

Reagan and Dodds (2016) performs lexicon based emotion analysis on literary texts to get emotional arcs. On clustering, 6 basic types of arcs are identified , concordant with Vonnegut (1995)'s shapes of stories. They use large texts and plot the general emotion over a 10000 word window of text using an emotion lexicon. The window size was chosen because dictionary based methods do not give good results for smaller sizes or for sentence level emotion as noted in Reagan A (2015) and Ribeiro FN (2016).

Alm (2008) studies sentence level emotion detection using lexicon , machine learning and combined approaches. It focuses on affect detection as expressed in the text, not reader emotion specifically. The linear classifier outperformed the lexicon based method and combination methods worked best. It should be noted that the machine learners were trained on the same dataset (children's fairy tales) and it may not generalize well to other genres.

2.5 Lexicons and other resources

A study Tabak and Evrim. (2016) compared ESN from S. Poria (2014), NRC emotion lexicon from Mohammad and Turney (2010), DepecheMood (DPM) from Staiano and Guerini (2014) and Topic-based DepecheMood (TDPM) from K. Song and Zhang (2016) on SemEval-2007 dataset and found NRC and DPM gave better performance on classification task. They also note that this result may vary across domains.

Wordnet from Fellbaum (1998) is a useful resource for NLP tasks. In WordNet, words are grouped by their meaning into sysnsets and synsets are connected to other semantically related synsets. Sentiwordnet, Baccianella and Sebastiani (2010), is a lexicon for opinion mining and assigns each synset in WordNet to either positive, negative or objective(neutral) opinion classes.

VerbNet, Schuler (2005), is an structured verb lexicon. Each verb is assigned to a verb class and arguments to the verb are assigned thematic roles, like actor and patient. It can be used to extract several useful details about the verb, including whether it is transitive and the syntactic structure indicating where the patient (the entity affected by the event caused by the verb) can be found in the sentence.

Framenet, Charles J. Fillmore and Lowe. (1998) is a lexicon created based on Frame semantics. It represents meanings of words in terms of semantic frames. For example,

words related to cooking are arranged around an Apply_Heat frame and roles (The Cook, the food, the container, etc) are labeled. It also holds relationships like inheritance between frames.

Deng (2017) is a study on entity/event level opinion mining and uses a +/- effects lexicon for the same. This is a lexicon of verbs marked with the state it imparts to its patient. For example, 'kill' is negative and 'feed' is postive. This goodFor/badFor lexicon was originally proposed in Goyal and III. (2010) . They try to automatically generate is a list of verbs , classified according to whether they are desirable or undesirable to its patient. A grammatical patient or theme as defined in memidex is the entity upon whom the action indicated by the verb is carried out. Goyal and III. (2010) concludes that their approach is not scalable. Choi and Wiebe (2014) is a later attempt to automatically generate a sense level lexicon of this kind, starting with an annotated seed lexicon. They get an f1 score of around 0.7 for both classes. This kind of lexicon can be potentially used as a heuristic to determine whether a particular event was good or bad for a particular character that appears as its patient.

2.6 Knowledge representation and Text mining in Stories

In Ludwig and Moens (2018), they present a way to extract action descriptions from narrative text. They use Genetic algorithms to resolve the ambiguity in results from NLP tasks and select the best representation from a number of states generated.

Lombardo and Pizzo. (2014) presents an ontology based tool for visualization of intentions that drive different actions of the characters in a drama. Character intentions are visualized as a tree and are automatically mapped to actions along the timeline of the drama. In this paper, they introduce an ontology called Drammar to represent dramatic elements like Scenes, plans, goals and agents. Their method depends on annotation provided by experts. Inclusion of the dramatic arc is mentioned as part of their future work. In his later work, Lombardo and Pizzo. (2015), he presents an agent based system with rules for prediction of emotional stages of characters. It relies on semantic representation of dramatic features.

Valls-Vargas and Zhu. (2014) proposes a system called Voz for automatic character identification. It first extracts entities and then generates feature vectors using several linguistic and external information to determine which entities are characters. In Valls-Vargas and Ontanon. (2017) Voz is used to identify narrative roles of characters from natural language stories. 87% of characters are identified and 68% of roles are assigned correctly. They also present a general methodology that can be used to analyse how error propagates in an arbitrary pipelined information extraction system.

Elsner (2015) tries to create an abstract representation of plot structures by generating topic models and character networks using co-frequency counts. Resulting model is able to distinguish genuine plots from randomly permuted text samples and beginnings from endings.

Chambers and Jurafsky. (2008) is a method to identify typical event chains from large volumes of text using unsupervised learning. Doust (2015) proposes a data structure called narrative threads to model event chains in stories. This is used for generation of suspenseful stories in this work, but he proposes using Chambers and Jurafsky. (2008) to generate event chains and using it as narrative threads. The work in Doust (2015) may then be adapted to model surprise, suspense and curiosity.

In a lot of works like Martin and Riedl (2017) natural language text is simply converted to formats similar to a subj-verb-obj representation to get structured data and used. Even though some information is lost and a lot of semantic details are not captured, this is straightforward to implement using out-of-the-box dependency parsers like StanfordNLP, Manning and McClosky. (2014).

Elson and McKeown. (2010b) generate social networks from 19th century british novels and studies the features of the resulting network. Their method depends on the ability to know when a conversation is happening between two characters. Nalisnick and Baird (2013) outlines a methods for extracting dynamic sentiment network among characters in Shakespeare's plays. The sentiment of a speech contributes to the edge weight between the speaker and the listener. Fernandez and Ulmer (2015) also builds sentiment networks for the purpose of antagonist and protagonist identification. They list verb phrases that connect two entities and calculate the sentiment using SentiWordNet, Baccianella and Sebastiani (2010) to determine their relationship.

2.7 Datasets

Gutenberg is large unannotated corpus of stories and has over 57000 texts. SemEval is a series of evaluations of semantic analysis systems, exploring the meaning in language. There are several datasets freely available under this for various natural language tasks. A system for human annotation of goal structures is presented in Elson and McKeown. (2010a) for Aesop-like stories. A corpus of short texts annotated by crowdsourcing is used in Li and Riedl. (2012) for learning event networks. Fast and Bernstein. (2016) uses stories from online writing community, Wattpad to mine human behaviour from fiction. Richardson and Renshaw. (2013) is a dataset of short stories published by Microsoft with associated questions for research in machine comprehension. babI Project from FaceBook also provides a number of datasets for NLP tasks.

Elson (2012) is a project that tries to encode narrative aspects of different genres of narratives, with notions of time, goal, plan, outcome, intention as well as affect.

There are very few story datasets annotated with reader emotion. Emobank, Buechel and Hahn (2017), contains many kinds of texts (7 of which are stories) annotated with writer and reader emotion in Valence-Arousal-Dominance format.

Alm (2008) provides an additional annotations for their dataset (not used in their thesis) of children's fairy tales where the 'feeler' of the emotion is also annotated. Some of these sentences are annotated with 'reader' as the 'feeler' and could be used for this project.

2.8 Intermediate steps

Out of the box methods exist for general NLP tasks such as Tokenization, Named Entity recognition, Co-reference resolution , Dependency Parsing, pos tagging and Sense Disambiguation. Sense disambiguation refers to recognizing which sense a word is used in, depending on the context.A dependency parser analyses the syntactic structure of a sentence and infers relationships between words. Named entity recognition is the process of identifying all named entities like people and organizations in the text. Co-reference resolution is the process of disambiguating different mentions of an entity and identifying which mentions refer to the same entity. Pos Tagging is the process of associating the words in the text with their corresponding part of speech. Tokenization is the process of identifying individual tokens in a sentence. These tasks form a pipeline, for example, you need to run tokenization and pos tagging before being able to run dependency parsing, and you need dependency parsing and named entity recognition to run co-reference resolution. SyntaxNet (2016) , StanfordNLP Manning and McClosky. (2014) , Spacy (2016) and NLTK Bird and Klein. (2009), are some of the most commonly used NLP libraries. SyntaxNet is a neural framework implemented using Tensorflow, Abadi et al. (2015). StanfordNLP is a framework supporting different types of models for many NLP tasks. It is implemented in Java, but has several wrappers like pycorenlp in python. NLTK is a widely used NLP library in python. It supports features like lemmatization, interface to many text corpora, WordNet, sense disambiguation but not features like co-reference resolution and dependency parsing. Spacy (2016) has a facts page showing tables for functionalities provided by each library and evaluations published in Honnibal and Johnson. (2015). Spacy 2.0 uses neural models for both for parsing and Named Entity recognition . Its comparisons with recently published (in 2017) systems are also provided. According to these evaluations, Spacy 2.0 exhibits good performance. Though it does not show as co-reference resolution as one of the features supported, there is a NeuralCoref pipeline extension for spacy 2.0 in GitHub that claims to be production-ready. Al Omran and Treude (2017) compares these libraries on several tasks (but mostly at the tokenization and pos tags level, and not more advanced

tasks like co-reference resolution) on Software documentation dataset and find that they have low agreement (only between 60-71%). They observe that StanfordNLP was most commonly used but Spacy performs better. They also note that while Spacy performs better in general, different libraries work better with different datasets.

Chapter 3

Methodology

3.1 Experiment Design

3.1.1 Emotion Model

For simplicity, only valence is considered. That is, classification involves only 2 discrete classes: positive and negative. Since only specific kinds of sentences are targeted, if prediction is not attempted , it is marked neutral. Note that neutral prediction is interpreted as "prediction not attempted" not as a classification into a "Neutral" class. Strength of the emotion is also not taken into account. However, the model can be adapted to account for intensity in the future.

3.1.2 Method

Machine learning methods would require datasets labeled with reader emotion that are large enough and cover different genres of stories. Procuring such a dataset is a problem. Unsupervised methods such as raKel from Ye and Xu (2012) seems to be a sensible option based on literature review , considering news stories are similar to fictional stories in many ways and existence of large unlabeled story datasets like [Gutenberg]. But a rule based method was chosen since it is more intuitive, and allows taking specific characteristics of stories into account explicitly.

3.1.3 Benchmark

Performance is compared against and in combination with Vader from Hutto and E.E. (2014), with the sentiment lexicon swapped out for the NRC emotion lexicon from Mohammad and Turney (2010) where 'joy' represents positive emotion and 'anger',

'fear' and 'sad' corresponds to negative emotion. The lexicon was swapped in order to adapt Vader, which is method for sentiment analysis, to a system for emotion analysis with minimum effort. Other emotion analysis techniques more sophisticated or with better accuracy might be more suitable for this problem, but these are not investigated in depth. Vader is a simple rule based method, and was developed with microblogging in mind, and not stories and for sentiment analysis, not reader emotion. Other rule based methods might be more suitable for stories, but Vader was chosen for pragmatic reasons, since it can be used out-of-the-box from NLTK. Since we are interested only in the validity of the rule itself, this should not matter. If it can improve on this simple method, it should be able to improve the more sophisticated methods as well.

3.1.4 The Rule

To generalize the situation outlines in the introduction, it can be said that the reader experiences happiness when good things happen to good characters but also when bad things happen to bad characters. Investigation is limited to this particular heuristic due to time constraints. Other heuristics may be used, (for example, when a character's plan fails) but they are not taken into consideration. This is a reasonable first step since it is comparatively easy to implement and can potentially affect a lot of sentences. Context, in this case, is limited to identifying if the characters are "good" or bad".

3.1.5 Target Sentences

Emotion detection is limited to events. For example, a sentence like *"He was happy."* explicitly expresses emotion but there is no event involved. In other words, nothing is "happening" to anybody here, so it need not get picked up by this rule.

3.1.6 Dataset

Emotions and inter character relations in a lot of stories are complex. But children's fairy tales from dataset Alm (2008) seems like a reasonable starting point as such emotional subtleties are not that common here. Sentences where emotions of the reader or main character are annotated are considered. There are 756 such sentences in total.

3.1.7 Emotion Detection - Design and implementation

In Doust (2015) importance of an event is formalized as the product of current level of empathy felt for a character and the perceived desirability of that event of that character. Hence, identifying the "good" and "bad" characters , and undesirability of an event for a particular character, could be combined to get the rule.

Out of the box methods or simplest methods are used for each of the steps in between and simplifying assumptions are made for generation of sentiment network and protagonist identification. Optimising the accuracy of each of these steps is beyond the scope of this project.

3.1.7.1 Proposed emotion detection algorithm

Algorithm 1:

1. Generate depparse and coref annotations using StanfordNLP.
2. Extract events for each sentence from the parse tree using 3.1.7.2.
3. Identify effect of each event using 3.1.7.4 and sum to get effect per sentence .

Based on literature review, Spacy (2016) seemed like a sensible option, but on implementing it, it was observed that the extension for coreference resolution, NeuralCoref, was not good at resolving pronouns inside quotations. Hence, StanfordNLP, Manning and McClosky. (2014) was chosen due to its popularity and its large number of features and options. It was implemented in python using pycorenlp. StanfordNLP uses a transition based dependency parser, (detailed explanation of which can be found in Nivre (2014)). It is powered by a neural network. It provides several options including use of word embeddings, training other models or using different pre-trained models, but default options were chosen and improving upon this is left to future work. Depparse requires sentence splitting, tokenization and pos tagging to be run beforehand. StanfordNLP's 'ssplit', 'tokenize' and 'pos' annotators were used with default options for these steps. For co-reference resolution, StanfordNLP offers options to use a deterministic, rule based annotator, or statistical or neural annotators. The deterministic annotator is faster but less accurate than the neural one. In this implementation, the neural annotator was used with default options.

3.1.7.2 Event Extraction

Algorithm 2:

1. Parse dependency tree to extract subj-verb-obj triples with an indication if the verb is negated or not.
2. Replace subj and obj with the resolved coreference mentions.
3. Replace verb with the disambiguated word sense.

Depparse outputs three kinds of dependencies - basic, uncollapsed dependencies and enhanced and enhanced++ dependencies with increased levels of enhancement. Dependencies from enhanced++ are used. Any type of verb is considered (A total of 6 types of verb tags are generated by pos tagging including present, past, plural, etc). It is arguable if this is the right approach, since the event "A will kill B" and "A killed B" may not evoke the same emotion but they both get translated to the same event A-kill-B. But both are clearly negative if B is the hero and positive if B is the villain. Given the scoping restrictions, since, intensity is not considered and fine grained emotions aren't targeted, all kinds of verbs are considered. All types of subjects except passive subject is taken as subject and all types of objects are taken as objects. For verbs with multiple subjects or objects, an event is generated per subject-object pair. If the token indicating the subject or the object is part of more than one reference chain, it is treated as having multiple subjects or objects as well and an event is created for each combination. For simplicity, only direct negation of the verb, that can be extracted from the dependencies is considered (like "A did not kill B"). Other cases, like, "He had no food", would still translate to He-have-food. Accounting for more types of negations is left for future work.

3.1.7.3 Sense disambiguation

Lesk algorithm from NLTK is used for word sense disambiguation. It implements Lesk (1986) which is a simple algorithm that associates the word with the sense whose description has maximum number of words in common with its context sentence. The most basic implementation is used for simplicity and improvements here is also left for future work.

3.1.7.4 Calculating event effect

Algorithm 3:

1. Look up event effect on object from +/- effects lexicon (+1, -1 or 0).
2. If the object is a "bad character" according to 3.1.7.5, negate the effect
3. If the object is "insignificant" according to 3.1.7.5, ignore the effect (effect is 0).

Modeling desirability of an event for a character is more challenging. Character goals in a story mostly govern which events are desirable. But this requires extensive intention modeling, world knowledge modeling and inference mechanisms. This is outside the scope of this project. +/- effects lexicon from Choi and Wiebe (2014) is used. They were originally developed for enhanced entity level sentiment detection but these could be made use of here. Detection of undesirability of an event to a character is limited by +/- effects. Cases where undesirability is not directly obvious from the verb (eg.

In a sentence "He left him a large treasure", the verb "left" alone does not impart any particular state. If the object were "a debt", it would have been negative but here, "treasure" makes it positive) or in cases where undesirability arises from other parts of the sentence, is not considered (for example, in "Her terror would not leave her."). +/- effects are defined for the grammatical patient of the verb. The patient is the object in most cases. In case of intransitive verbs, sometimes, the subject becomes the patient. For example, in "He died", "he" is the patient. In this implementation, only verbs with objects are considered for simplicity.

3.1.7.5 Character classification

Algorithm 4:

1. Build sentiment network among characters using 3.1.7.6.
2. Identify the protagonist using 3.1.7.7.
3. If a character is negatively related to the protagonist, he/she is bad.
4, If coref chain doesnot exists or if none of the mentions in the coref chain indicates the entity as animate, it is considered insignificant.

Sentiment networks and protagonist identification are used to identify "good" and "bad" characters. An assumption is made that all characters negatively related to the protagonist are bad. Anyone positively related or not related is good. This seems reasonable for children's fairy tales, but it does not account for more complicated personality types and relationships in other literary works where the line between good and bad isn't as clear and reader empathy may vary or is more subtle. Character detection is researched in Valls-Vargas and Zhu. (2014) . In Fernandez and Ulmer (2015) they add additional heuristics like if the entity is animate. Character detection in children's fairy tales is more complicated since many characters are not even named (simply refered to as "the step mother", for example) , they are sometimes animals or trees. Character detection is not focused in this project. The simplest assumption that any entity with more than one mention and is identified as animate at least once in the coref chain is a character, is made and applied. More sophistication is left for future work.

3.1.7.6 Building Sentiment Network

The network is not dynamic. Therefore, some issues are expected for instances like if characters that are initially good later turns bad. Most works covered in literature review obtain the sentiment network by combining the sentiment of all sentences where two characters co-occur. However, in this dataset, since there are many instances where positively related characters experience sadness (a negative sentiment) together in the

Algorithm 5:

1. **for** all pairs of entities **do**:
2. Get all events where one entity is the subject and the other is the object.
3. Sentiment between them is the sum of +/- effects of verbs in all the events.
4. **end for**

same sentence, this gave confusing results. Hence, a this approach of extracting only the verbs (similar to that in Fernandez and Ulmer (2015)) was chosen.Instead of using SentiWordNet as in Fernandez and Ulmer (2015), the +/- effects lexicon was used for determining sentiment since this is more intuitive for verbs.Though model can easily be adapted to generate a dynamic network, (by generating the network after each sentence is read), it is left out for simplicity. If a +/- lexicon with intensity scores is available (or by using SentiWordNet), the model can be adapted to have edge weights as well (for a scenario where emotion intensity is also being calculated).

3.1.7.7 Identifying Protagonist

Algorithm 6:

1. Extract all events using 3.1.7.2.
2. Count number of times each entity appears as subject.
3. Entity appearing as the subject the most is the protagonist.

An assumption is made that the entity appearing as the subject the most is the protagonist. The entity with most mentions or entity appearing most as either subject or object are also arguably reasonable assumptions, but this one was chosen since it seemed to give best results for this dataset. This was not deeply investigated since it is not the focus of this study. Some stories have more than one protagonist , eg Hansel and Gretel, but this model does not allow for multiple protagonists. However, as long as they are positively related (or not related at all), they will both be classified as good characters and we won't have an issue with emotion detection, so this is not a problem.

3.2 Demonstration

To demonstrate the working of the algorithm, an example is shown with a small story of 5 sentences:

Hansel and Gretel lived near the woods with their step-mother, Lucy. Once, when food

ran low, Lucy abandoned them in the forest. There, they met Wendy the witch. Wendy imprisoned them. When they got a chance, they killed her and escaped.

The co-reference chains and dependency parses identified by standfordNLP are attached in the zip file. The result of coreference resolution is visualized in Figure 3.1 highlighted in different shapes and colors.

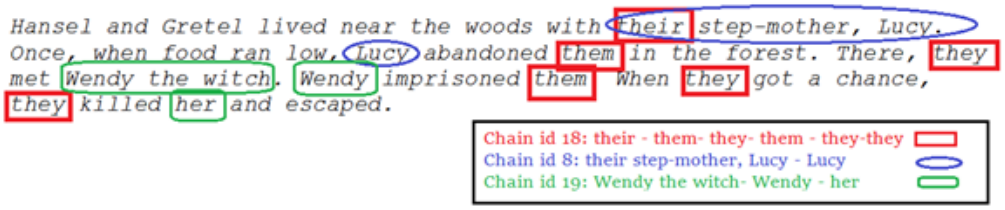

FIGURE 3.1: Coreference chains

Coreference seems almost perfect except that Hansel and Gretel were not associated to the rest of its chain. Events extracted for this text and emotion predicted for each is shown in Table 3.1.

Verb	Subject	Object	Negation	Verb Effect	Event Effect
Synset('live.v.07')	Hansel	-	False	1	0
Synset('live.v.07')	Gretel	-	False	1	0
Synset('run.v.33')	food	-	False	0	0
Synset('abandon.v.05')	8	18	False	-1	-1
Synset('meet.v.09')	18	19	False	1	-1
Synset('imprison.v.02')	19	18	False	-1	-1
Synset('grow.v.08')	18	chance	False	0	0
Synset('kill.v.15')	18	19	False	-1	1
Synset('scat.v.01')	18	18	False	-1	0

TABLE 3.1: Table showing events extracted and event effects predicted

As these sentences are fairly simple, depparse works well and all expected events are retrieved. Coreferring mentions are represented by their chain id. The activity counts (number of times, an entity occurs as the subject) for each entity is shown in Table 3.2

The protagonist identified is "they" since it has the maximum occurrence as subject. All mentions without a coreference chain, - Hansel, Gretel, food and chance are considered

Coref Chain Id	Character	Number of appearences as subject
8	Lucy	1
18	they (Hansel and Gretel)	4
19	Wendy	1

TABLE 3.2: Table showing coref chains and activity counts

insignificant. The sentiment network generated from this set of events is visualized in Figure 3.2. The insignificant entities are coloured grey, good characters and relationships are green and bad characters and relationships are red. The interactions between the characters (verbs of which they appear as subject and object) are labeled on the edges. Since Hansel and Gretel were not associated to the rest of their co-reference chain, "they" appears as a single entity tha is separate from both. This could potentially cause errors, but in this particular case, this error does not effect stages down the pipeline. This is studied in more detail in the next chapter. The ideal network is shown in Figure 3.3.

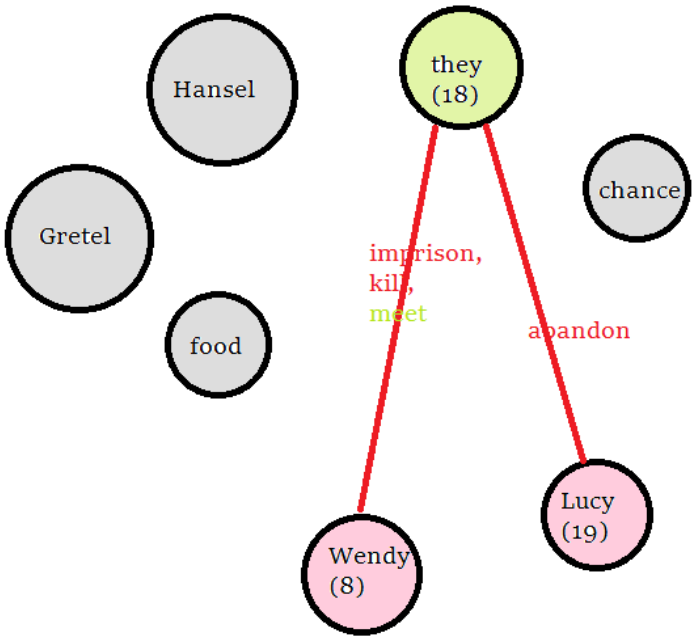

FIGURE 3.2: Sentiment network generated

The original output as a python dictionary can be found in the zip file. Wendy is identified bad because she is negatively related to the protagonist due to the actions – "imprison" and "kill" and Lucy is identified as negative due to "abandon". This character classification combined with verb effects from +/- effects lexicon as per Choi and Wiebe (2014) ,gives the event effect as per 3.1.7.1. Both the verb effect and event effect is shown in Table 3.1.

Note that in the event – 18 kill 19, even though "kill" has a negative effect, the event has a positive effect as 19 is negatively related to the protagonist, 18. Vader predicts wrongly for this instance in table 3.3 that shows final predictions for each sentence from each of the systems.

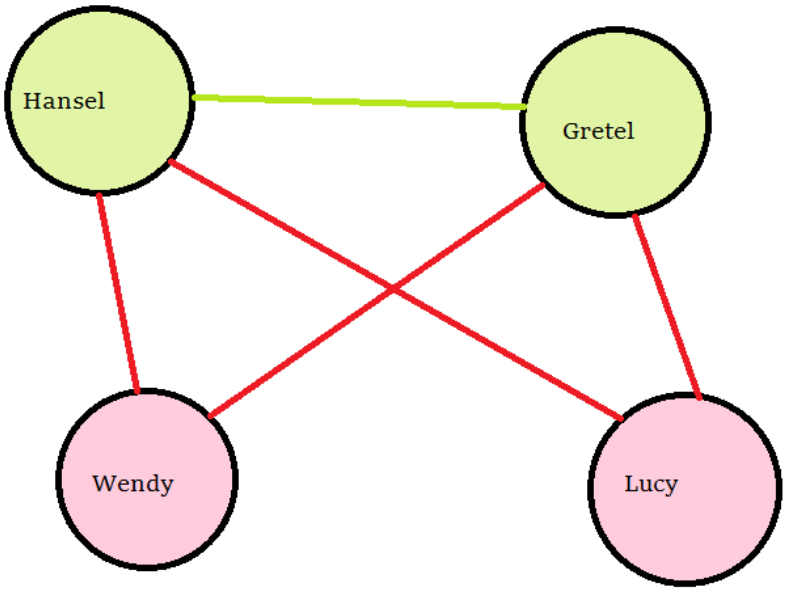

FIGURE 3.3: The ideal Sentiment network

id	rawtext	class	A	B	C
1	Hansel and Gretel lived near the woods with their step-mother, Lucy.	N	N	N	N
2	Once, when food ran low, the Lucy abandoned them in the forest.	-	-	-	-
3	There, they met Wendy the witch.	N	-	-	-
4	Wendy imprisoned them.	-	-	-	-
5	When they got a chance, they killed her and escaped.	+	+	-	+

TABLE 3.3: Table showing results from proposed algorithm(A), vader(B) and combined (C)

3.3 Evaluation

3.3.1 Emotion detection algorithms compared

- A - The new algorithm formed by combination of sentiment network and +/- effects.
- B - Simple emotion detection using vader (code was changed to use NRC emotion lexicon instead of lexicon for sentiment analysis)
- C - A combined approach where emotion from A is taken if prediction is attempted (not neutral), the emotion predicted by vader is used, otherwise.

3.3.2 Question

Can NLP techniques be used to introduce heuristic rules specific to the nature of stories to improve reader emotion detection?
The heuristic experimented with here is: Readers experience positive emotion when good things happen to good characters and bad things happen to bad characters.

3.3.3 Hypothesis

The scoping restrictions imply that this particular approach (Algorithm A) may not be effective for all types of sentences. Additionally, since the proposed method is heavily dependent on lexicon based methods, high recall is not expected. Instead, high precision is expected. Its combination with the lexicon based method (C) is expected to give better f1 score than either A or B.

3.3.4 Method

The example shown above is a very simple case. It is short, the sentences are not complex, all the characters are named and it was written specifically to show the working of this method. Its performance on the real dataset is not likely to be this good. Evaluation is carried out by generating error metrics over the results and performing in depth analysis of one story that exhibits closest accuracy measures to the mean case.

3.3.4.1 Metrics

Accuracy, precision, f1 , recall of each algorithm is drawn and compared. The instances where wrong classification was made are studied manually and the reason for failure is determined.

3.3.4.2 Mean story

In real stories, greater error is expected in the intermediate steps and it is likely to propagate into the final results. Since the results expected from each of the intermediate steps is not annotated, error at each step and its propagation cannot be studied computationally. It is hard to follow each step of the algorithm manually as above for all 176 stories that are considerably longer. Hence one story that exhibits metrics close to the mean values is selected and the performance of the algorithm on this story is studied in depth to determine what went well and what went wrong.

Chapter 4

Results

4.1 Error Metrics

The dataset consists of 176 stories containing a total of 15379 sentences. Only 886 of them are marked with reader or main character emotion. The sentence ids in the jsons from StanfordNLP parses vary in some cases with the sentence ids annotated in the dataset due to errors in sentence recognition. Basic check was done for similarity for each sentence in the dataset with reconstructed sentences from the jsons and sentence ids were corrected. Sentences that could not be matched were discarded. Sentences marked neutral were also discarded.
Remaining number of valid sentences for evaluation was 748.
Out of these predictions were attempted for 69.

The accuracy for each algorithm calculated as :
$accuracy = correctPredictions/totalPredictions$
where total predictions = 69

The result is shown in Table 4.1

Algorithm	Accuracy
A	0.5972
B	0.7079
C	0.696

TABLE 4.1: Accuracy for each system

Accuracy for C is less than B which implies that accuracy drops slightly when A is combined with B, instead of getting better. Precision, recall and F1 score for each class is shown in Tables 4.2 and 4.3.

Algorithm	Precision	Recall	F1
A	0.421	0.0625	0.054421
B	0.5683	0.7304	0.3196581196
C	0.5588	0.7421	0.31879

TABLE 4.2: Error metrics for + class

Algorithm	Precision	Recall	F1
A	0.7741	0.049	0.0463320
B	0.8849	0.4579	0.301759133
C	0.874	0.4558	0.2995951

TABLE 4.3: Error metrics for - class

4.2 Analysis

Precision of A is not as high as expected. As a result, incorporating it into vader causes it to perform worse, not better, except for the slight increase in recall for + class. Out of the 69 predictions, 40 were correct and 29 were wrong. Out of the correct instances, 11 were instances where vader had made the wrong prediction and 4 were instances where vader had made no prediction. However, 13 of the mistakes were cases where vader was right in the first place and 9 in cases vader had only predicted "neutral" but A made the wrong prediction. To see why this is happening, the mistakes made by A were analysed manually. The reasons for failure are attributed to the following reasons. The number of instances exhibiting each error is added alongside:

- Error in +/- effects lexicon - 15
- Sense disambiguation - 12
- Pos Tagging - 1
- Character Classification - 4
- Protagonist Identification - 1
- Co-reference resolution - 3
- Mixed events - 3
- Ambiguous emotion - 2
- Limitations – scope of design , scope of implementation - 1
- Indirect negation and conditionals - 5

While some sentences are affected only by one of these errors, some are affected by a combination of many of them. Figure 4.1 shows different components of the system and

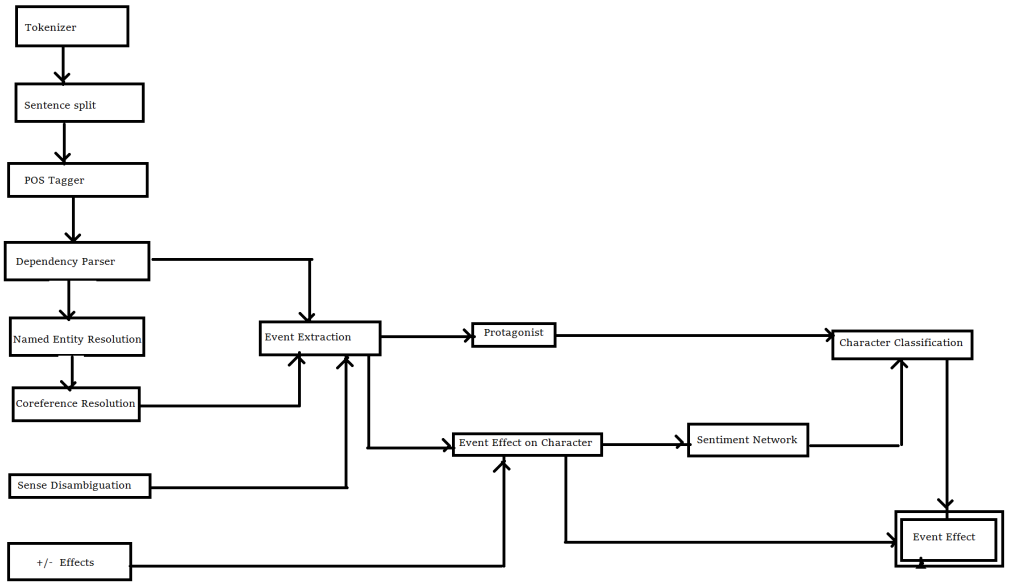

FIGURE 4.1: Dependency pipeline

how they depend on each other, forming a pipeline similar to that studied in Valls-Vargas and Ontanon. (2017). There are two kinds or effects errors can have: direct and indirect effect. Direct effect is when simply correcting the error at this stage will give the correct result. Correct classification in each of these instances in this context is considered to be either the correct class or neutral. This could be because after correcting this component leads to any of the following conditions:

- There are no more errors.
- There are errors earlier up in the pipeline, but they are not reflected down past this stage.
- Errors along the pipeline cancel each other out by chance, giving the correct result out of luck.

Indirect causes influence the result, but the error is itself is caused by a combination of several of these factors. Even if this error is corrected, the system will not function as expected. This may because either of the two reasons:

- There are more errors down the pipeline, that will still lead to erroneous result. Or,
- This error is caused by errors earlier up the pipeline and that is what needs to be corrected.

4.3 Illustration

Complete, sentence-wise analysis of all the mistakes can be found in the zip file, but an example of correct classification and examples for different types of incorrect classifications are provided to here for illustration. Each of these sentences are taken from Alm (2008) dataset and all definitions of synsets are taken from WordNet.

> *"Besides that, the sisters plagued her in all sorts of ways, and laughed at her"*

This sentence was classified positive by vader, probably due to the word, laughed, but A correctly classifies it as negative since the object of "plagued" is a good character. Even though we get the correct result, some errors and limitations still exist:

1. Even though "her" is a good character, she was not correctly identified as the protagonist due to error in co-reference resolution. But note that for the purposes of emotion detection, this has been sufficient. This suggests that perfect accuracy at these levels might not be necessary.
2. Ideally we would like the verb "laugh" be positive and be associated with "the sisters", who should be classified as bad characters and hence the event would be negative. But "laugh" has 0 effect in the lexicon, since even though it depicts happiness in the person who performed the action, the verb, by itself, cannot be generally classified as either good for or bad for any patient. So "laugh" event gets 0 effect and not negative.
3. The error in 1 does propagate to the character classification level. The sisters are not classified as bad, since, and according to this implementation only characters performing harmful actions on the protagonist is classified bad, not characters performing bad actions on any good character. The error did not reflect anywhere because there were no more +/-events featuring the sisters as objects.

> *"They lived together very happily, and the queen had a son."*

'Had' was disambiguated to Suffer.v.2 whose definition is (undergo (as of injuries and illnesses)) instead of have.v.17 (give birth to). Have.v.17 has + effect since the event brings the object, "son" into existence and since son identified a good character, his coming into existence is a good thing and the event has a positive effect. Even though "they lived together happily" is a stronger indicator of the positive nature of the sentence, it is outside the scope of this rule.

> *"... Anne Lisbeth murmured these words to herself.."*

"These words" is brought into existence by the action "murmur", so it is considered to have a positive effect on the patient, "these words", but since "words" is not a significant character, it should have been ignored while sentence effect was calculated. However, "words" was not correctly identified as insignificant. This is because of an error in co-reference resolution where two mentions of "they" were assigned to this chain incorrectly. These mentions are animate and since animate entities are considered significant, so was this.

> *"The wolf had the fox with him, and whatsoever the wolf wished, that the fox was compelled to do, for he was the weaker, and he would gladly have been rid of his master."*

In this story, wolf has the most appearances as subject – 31 and the fox has a close -29 and they are negatively related. Since fox is the clever one in the story, the readers are more likely to root for the fox, hence the protagonist is identified wrong. The error however, is caused by the incorrect disambiguation of the verb "had" which was disambiguated to suffer.v.2 instead of have.v.1. have 1 is not part of the lexicon and would have received 0 effect. The negative feeling is induced by the fox being "compelled" to do things for the wolf. The current implementation did not consider events without subjects, so this event was missed. The results with this corrected are shown in tables 4.4 and 4.5. Accuracy improved to 0.62, precision and recall of both classes also show improvement and predictions are now attempted for 110 instances. However, the synsets for 'compel' are not present in the lexicon, and this sentence is still not classified negative.

Algorithm	Precision	Recall	F1
A	0.46666666666	0.109375	0.08860759493670
B	0.568389057750	0.7304687	0.3196581196
C	0.5523255	0.7421875	0.31666666666

TABLE 4.4: Errors metrics: + class after including passive subject as object

Algorithm	Precision	Recall	F1
A	0.82	0.0841889117	0.0763500931098
B	0.884920634920634	0.4579055441	0.3017591339
C	0.874015748031	0.45585215605	0.29959514170040

TABLE 4.5: Errors metrics: - class after including passive subject as object

> *"The warmth revived the poor little creature; but when the children wanted to play with him, the duckling thought they would do him some harm; so he started up in terror, fluttered into the milk-pan, and splashed the milk about the room."*

There are multiple events in this sentence, some of which are good, others are bad. "Revive" is correctly identified as positive but since the other events (that are negative) are outside the scope of the rule, they do not get picked up.

> *"And with this thought he consoled himself a little."*

The emotion induced by this sentence is ambiguous in the given model. This is a limitation of the emotion model used. It is a little more positive than the presumably unfortunate events before this, but is still not entirely positive. It was marked negative by annotators but classified positive since consoling has a positive effect on "him", the protagonist.

> *"How the sunshine cheers me, and how sweet and refreshing is the rain; my happiness overpowers me, no one in the world can feel happier than I am."*

Though being overwhelmed is bad, being overpowered by happiness is good. This is a limitation of the design. The event with cheer is missing because cheers was tagged as noun. But cheer is missing from the lexicon as well, and hence would not have received + effect as intended.

> *"Listen to the story of Jemima Puddle-duck, who was annoyed because the farmer's wife would not let her hatch her own eggs."*

The negation of hatch is indirect. This implementation only takes direct negations of verbs into account (for example, "did not hatch"). Furthermore, eggs are not considered insignificant, since it is marked animate in one instance.

> *"Still, as it grew, it complained, "Oh! how I wish I were as tall as the other trees, then I would spread out my branches on every side, and my top would over-look the wide world."*

This was predicted positive because spreading out is considered to have a positive effect on branches and branches are not insignificant (correctly so, in this story where the protagonist is a tree). However, the event "spread out" never occurred, it is just wistful thinking by the tree and that is what makes the emotion negative. This has not been taken into account in the implementation.

4.4 Mean Story

The conditions under which an error does or does not propagate to the final stage, and to what extent is hard to generalize without conducting similar manual analysis for all sentences. Additionally, predictions have not been attempted for a large number of sentences. Some may be for valid reasons that it is out of the scope of this rule, but manually verifying nearly 700 sentences in the given time frame is not practical. Hence, a story near with mean accuracy and prediction ratio has been selected and studied to give some understanding as to what might be going on. The accuracy and ratio of predictions made for each story was calculated and the mean was taken. The distance of each story from this mean was calculated as sum of squares of difference from the mean value.

$distance_{story} = (accuracy_{story} – accuracy_{mean})^2 + (predRatio_{story} – predRatio_{mean})^2$
The story with minimum distance was selected.
The mean of story-wise accuracy is: 0.592
Story-wise mean ratio of predictions made is: 0.127
Story with minimum distance : Clever-elsie

4.4.1 Missed Predictions

Complete sentence-wise analysis can be found in the appendix. An overview is presented here with some examples. There are 24 valid sentences that were identified neutral, 22 of which are out of scope for this rule for one or more of the following reasons:

- 8 sentences had explicit statement of emotion as in *".. she was alarmed.."* or expression of it *"....she sat and wept and screamed with all the strength of her body.."* that are not positive or negative effect events.
- In 14 sentences, the reason for undesirability of the event is not directly obvious from the verb. It requires some inference. For example, *"... saw a pick-axe exactly above her.."* is negative because of the looming danger of the pick axe falling and hurting her.
- Others were missed because events in passive voice were missed as noted and corrected earlier, or because of the indirect negation, also discussed earlier.

4.4.2 Error Propagation

We would expect error at each step to accumulate. So we would expect error at the top most parts of the pipeline to have the most effect. However , it is interesting to

note from table that most errors are from Sense disambiguation and the +/- lexicon which are not affected by other parts of the pipeline. In 17 of these cases (more than 50% of total errors), errors could have avoided simply by getting these two steps right. There are 286 co-referring mentions, 1744 dependencies in the parse tree , 1632 pos-tags and 194 events . Therefore analysing errors and causality at these stages manually is time consuming. The interdependencies and evaluation methodology are only briefly discussed here, but the actual evaluation is left for future work. The directed acyclic graph as proposed in Valls-Vargas and Ontanon. (2017) is shown in Figure 4.1 . Direct links indicate that error can be passed down directly along this edge. To study how the error propagates, individual errors at each step have to be calculated, as well as what percentage of these errors are caused by each of its parents and how much new error is introduced in that stage.

A gold lexicon is available with Choi and Wiebe (2014). If this is used instead of the generated lexicon, even though many more sentences will be missed, there will not be any error from the lexicon. The result after substituting the lexicon is shown in tables 4.6 and 4.7

Algorithm	Precision	Recall	F1
A	0.2	0.00390625	0.0038314176245210726
B	0.5683890577507599	0.73046875	0.3196581196581197
C	0.5696969696969697	0.734375	0.32081911262798635

TABLE 4.6: Error metrics for + class with gold lexicon

Algorithm	Precision	Recall	F1
A	1.0	0.012320328542094456	0.012170385395537527
B	0.8849206349206349	0.45790554414784396	0.301759133964817
C	0.8893280632411067	0.4620123203285421	0.30405405405405406

TABLE 4.7: Error metrics for - class with gold lexicon

Only 12 predictions were made but C now outperforms B in all cases. This suggests that if error at the intermediate steps can be reduced, then this rule is effective, and that the undesirability and sense disambiguation steps could be good places to start.

Chapter 5

Conclusions

5.1 Summary and Discussion

The work presented here puts forward an idea for reader emotion detection and a way to implement it using popular NLP techniques. Some compromises were made due to lack of time, as noted earlier, and has been left for future work. While some more work needs to be done before claims can be made about the initial research question and the validity of the rule conclusively, this work gives insight into its potential as well as challenges that arise. The system succeeds in getting a slightly better performance than vader when a gold lexicon is used for +/- effects, indicating that if error of intermediate steps can be reduced and the lexicon is expanded for better coverage, incorporation of this rule can improve emotion detection.

The work presented here shows that while including a rule such as the one proposed here, might be helpful in reader emotion prediction, expected results cannot be achieved unless better accuracy can be achieved in the intermediate steps. In many cases, the error is a result of a combination of many factors. Using a combination of techniques that affect each other leads to a condition where error from one step propagates to the next, causing the algorithm to behave differently than expected. Further, it indicates that improving accuracy at some steps might have greater impact than others. Errors in some steps (for example, in coreference resolution) can be tolerated. The work suggests that improving sense disambiguation and +/- effects lexicon (by which most errors are caused) will have the most effect on emotion detection.

5.2 Limitations and Future Work

This section summarises limitations of this project, briefly outlines a plan for addressing them and details more directions for future work.

- Due to the extent of manual work required to complete the evaluation, only a few examples were studied. It provides some idea as to what the challenges are and why they arise, but to definitively know how errors in different components effect the final result, a more systematic and thorough evaluation is required. Study of error propagation is also restricted by the fact that there are only a few sentences annotated by reader emotion per story. The work done here seems to suggest that Sense disambiguation and Undesirability has the most effect, but by figuring out the individual errors at each step and the percentage of errors caused by the previous steps, the component affecting the result the most can be identified definitively and concentrated on. It can also tell us what threshold of accuracy is required at each step to see a substantial improvement in emotion detection.

- The work presented here uses the most basic off the shelf methods for the intermediate steps. Once the most important components are identified as described above, work can be done to improve its accuracy. As noted in Al Omran and Treude (2017), different libraries work better with different datasets. The library most suited for stories can be identified and used. Lesk (1986) is a very basic algorithm put forward in 1986 and much research has happened in this area since. Such methods can be investigated.

- Alternatively, the system can be redesigned to use fewer layers. For example, character classification to good and bad can be tried out using entity level sentiment detection techniques as in Deng (2017). This might work well in this dataset since the antagonist is very often described by adjectives like "evil step mother". This way the sentiment network and protagonist identification layers can be removed.

- The patient of a verb is assumed to be the object. The small change of taking the passive subject as an object as discussed earlier led to 40 more predictions and improved accuracy to 0.62. Sentence wise analysis of the result after this was not done due to time constraints. Verbnet can be used to identify the patient more correctly since it has information on whether a verb is transitive or intransitive and how the agent and patient are usually arranged around the verb syntactically.

- The rule was only evaluated on the dataset of children's fairy tales. Its performance on other types of stories should be studied. Children's fairy tales are simple, have simple inter character relationships and emotions and has a more event based structure. Most other literary works have a lot of dialogues, subtler relationships and emotions and is likely to present new challenges.

- This rule covers only a small set of sentences and are centered around the verbs. For a complete system, more heuristics should be developed. Research on associating opinion words to entities can be extrapolated to associating emotion words with entities. This way coverage can be extended to nouns , adjectives and adverbs.

- Rule based method was chosen because of simplicity and interest. Unsupervised methods like Ye and Xu (2012) can be tested and compared on this dataset. Additionally including semantic information in the preprocssing step or as additional features can be experimented with. In a pre-processing step, coreference resolution and character classification can be done and entities can be replaced with the corresponding archetypes. Dependency parses can be passed as additional features in the hope the system may pick up associations of emotion words with different entities.

- Aruguably, reader emotion might not be best expressed by +/-. The model can be adapted to have an associated intensity if a +/- effects lexicon marked with intensity can be obtained. Additionally the sentiment networks could have edge weights indicating strength. Still, emotions like suspense are neither positive nor negative but are an important component of the experience of reading. In the original goal of calculating dramatic effect, suspense is considered as a different component, not part of emotion. Doust (2015) proposes a method to generate suspenseful stories using a data structure called Narrative thread. A structure similar to this can be obtained using the event chain extraction method described in Chambers and Jurafsky. (2008) and could potentially be used to model suspense, surprise and curiosity.

- Many errors were because the system was not able to pick up indirect negations. Much research exists exploring this issue. This could be investigated to improve the system.

- Some errors were because insiginificant characters were not identified correctly. In this implementation if any one mention is marked animate, the chain is considered animate. In Labiba Jahan and Finlayson. (2017) animacy is calculated by a vote over all mentions in the chain. This might help solve this problem.

- The sentiment network is taken as static in this implementation. It could be made dynamic by creating the network at each step (when a new sentence is read) by using only the text read so far and giving weightage to the more recent actions. This might be a more accurate representation of the reading experience and can pick up when a character exhibits traits like betrayal. This was avoided to save computational time.

- Bad characters are assumed to be those who does harmful action to the protagonist. Actions against other good characters is not considered. This might be a problem in stories from other genres. In such cases, it might be interesting to use network theory to study propagation of sentiment along the edges of the network.

5.3 Reflection

The project work stuck to the schedule for the most part, but the topic changed a lot during the course of the project as a result of scoping down. A clear decision on a simple topic from the beginning would have helped the project be better focused and organized. For example, the literature review is broader than it is deep. Complete landscape of emotion analysis and of text mining not captured. The heuristic was picked on the basis that lexical methods would fail on a particular type of sentence. Though this is a reasonable choice, ideally, performance of state of the art systems that show good performance on text, news stories in particular should have been tested on this dataset. A study of what type of sentences were misclassified could have helped guide selection of the heuristic. A heuristic addressing the class of sentences with most mis-classifications could have been developed. Implementation and testing also was completed in time except for one issue that was noticed only during analysis or results (the passive subject was not taken as a patient). Ideally all analysis should have been done after the change was made, but this was discovered after a majority of the analysis was done so the analysis is reported as is.

Bibliography

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

H. Porter. Abbott. The cambridge introduction to narrative., 2008.

An . Agrawal. Unsupervised emotion detection from text using semantic and syntactic relations. in: Proceedings of the ieee/wic/acm international joint conferences on web intelligence and intelligent agent technology, pp 346–353., 2012.

Fouad Nasser A. Al Omran and Christoph Treude. Proceedings of the 14th international conference on mining software repositories, pp. 187-197. ieee press, 2017.

Ebba Cecilia Ovesdotter Alm. Affect in* text and speech. university of illinois at urbana-champaign, 2008.

Facebook babI Project. Facebook babi project.

Esuli A. Baccianella, S. and F. Sebastiani. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining., 2010.

Anupam Basu Bhowmick, Plaban Kumar and Pabitra Mitra. Classifying emotion in news sentences: When machine classification meets human classification., 2010.

Potdar V Binali H, Wu C. Computational approaches for emotion detection in text. in: Ieee international conference on digital ecosystems and technologies (dest), pp 172–177, 2010.

Edward Loper Bird, Steven and Ewan Klein. Natural language processing with python. o'reilly media inc., 2009.

Sven Buechel and Udo Hahn. Emobank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis., 2017.

Martinez-Barco P Canales L. Emotion detection from text: a survey. in: Processing in the 5th information systems research working days (jisic 2014), pp 37–43, 2014.

Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative event chains., 2008.

Collin F. Charles J. Fillmore, Baker and John B. Lowe. The berkeley framenet project., 1998.

Yoonjung Choi and Janyce Wiebe. +/-effectwordnet: Sense-level lexicon acquisition for opinion inference, 2014.

Chopade CR. Text based emotion recognition: a survey. int j sci res (ijsr) 4(6):409–414, 2015.

Lingjia. Deng. Entity/event-level sentiment detection and inference., 2017.

R. A. Doust. A domain-independent model of suspense in narrative (doctoral dissertation, the open university), 2015.

Paul. Ekman. "an argument for basic emotions". cognition and emotion., 1992.

Micha. Elsner. Abstract representations of plot struture., 2015.

David. Elson. Dramabank: Annotating agency in narrative discourse., 2012.

David K. Elson and Kathleen R. McKeown. Building a bank of semantically encoded narratives, 2010a.

Nicholas Dames Elson, David K. and Kathleen R. McKeown. Extracting social networks from literary fiction., 2010b.

William McGrath Pranav Rajpurkar Fast, Ethan and Michael S. Bernstein. Augur: Mining human behaviors from fiction to power interactive systems., 2016.

Christiane Fellbaum. Wordnet: An electronic lexical database., 1998.

Michael Peterson Fernandez, Matt and Ben Ulmer. Extracting social network from literature to predict antagonist and protagonist., 2015.

Gustav Freytag. Freytag's technique of the drama, an exposition of dramatic composition and art by dr. gustav freytag: An authorized translation from the sixth german edition by elias j. macewan, m.a. (3rd ed.), chicago: Scott, foresman and company, lccn 13-283,copyright 1894, 1900.

Ellen Riloff Goyal, Amit and Hal Daumé III. Automatically producing plot unit representations for narrative text., 2010.

Verma A Grover S. Design for emotion detection of punjabi text using hybrid approach. in: International conference on inventive computation technologies (icict), vol 2, pp 1–6, 2016.

Project Gutenberg. Project gutenberg. (n.d.).

Matthew Honnibal and Mark Johnson. An improved non-monotonic transition system for dependency parsing., 2015.

C.J. Hutto and Gilbert E.E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. eighth international conference on weblogs and social media (icwsm-14). ann arbor, mi, june 2014, 2014.

L. Chen S. Feng D. Wang K. Song, W. Gao and C. Zhang. Build emotion lexicon from the mood of crowd via topic-assisted joint nonnegative matrix factorization, 2016.

Geeticka Chauhan Labiba Jahan and Mark Finlayson. Building on word animacy to determine coreference chain animacy in cultural narratives., 2017.

M. Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone., 1986.

Stephen Lee-Urban Darren Scott Appling Li, Boyang and Mark O Riedl. Crowdsourcing narrative intelligence., 2012.

Xu Li. Text-based emotion classification using emotion cause extraction. expert syst appl 41(4):1742–1749 ., 2014.

Changhua Yang Lin, Kevin Hsin-Yih and Hsin-Hsi Chen. Emotion classification of online news articles from the reader's perspective., 2008.

Rossana Damiano Cristina Battaglino Lombardo, Vincenzo and Antonio Pizzo. Automatic annotation of characters' emotions in stories., 2015.

Vincenzo Lombardo and Antonio Pizzo. Ontology–based visualization of characters' intentions., 2014.

Quynh Ngoc Thi Do Cameron Smith Marc Cavazza Ludwig, Oswaldo and Marie-Francine Moens. Learning to extract action descriptions from narrative text., 2018.

Mihai Surdeanu John Bauer Jenny Finkel Steven J. Bethard Manning, Christopher D. and David McClosky. The stanford corenlp natural language processing toolkit, 2014.

Jodok Vieli Wojciech Witoń Rushit Sanghrajka Daniel Inversini Diana Wotruba Isabel Simo Sasha Schriber Mubbasir Kapadia Marti, Marcel and Markus Gross. "cardinal: Computer assisted authoring of movie scripts." in 23rd international conference on intelligent user interfaces, pp. 509-519. acm, 2018.

Prithviraj Ammanabrolu Xinyu Wang William Hancock Shruti Singh Brent Harrison Martin, Lara J. and Mark O. Riedl. Event representations for automated story generation with deep neural nets., 2017.

Michael Mateas and Andrew Stern. Façade: An experiment in building a fully-realized interactive drama, 2003.

Mehrabian. Basic dimensions for a general psychological theory. pp. 39–53., 1980.

memidex. memidex, retreived 03 september, 2018.

Howard Packer Millard, West-Taylor. The ideal readerbot: Machine readers and narrative analytics, 2018.

Howard Yvonne Packer Millard, Hargood. The storyplaces authoring tool: Pattern centric authoring. in authoring for interactive storytelling 2017. 6 pp., 2017.

Saif Mohammad and Peter Turney. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon, 2010.

Eric T. Nalisnick and Henry S. Baird. Extracting sentiment networks from shakespeare's plays., 2013.

NeuralCoref. Neuralcoref.

Nivre. Joakim nivre's eacl 2014 tutorial, 2014.

collins Ortony, Clore. " the cognitive structure of emotions (1988)., 1988.

pycorenlp. pycorenlp.

Mitchell L. Kiley D. Danforth C.M. Reagan, A.J. and P.S. Dodds. The emotional arcs of stories are dominated by six basic shapes., 2016.

Williams JR Danforth CM Dodds PS Reagan A, Tivnan B. Benchmarking sentiment analysis methods for large-scale texts: a case for using continuum-scored words and word shift graphs, 2015.

Gonçalves P Gonçalves MA Benevenuto F Ribeiro FN, Araújo M. Sentibench - a benchmark comparison of state-of-the-practice sentiment analysis methods., 2016.

Christopher JC Burges Richardson, Matthew and Erin Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text., 2013.

E. Cambria A Hussain G. B. Huang S. Poria, A Gelbukh. Emosenticspace: A novel framework for affective common-sense reasoning,., 2014.

Manmeet Dhaliwal Jon Rokne Sailunaz, Kashfia and Reda Alhajj. "emotion detection from text and speech: a survey." social network analysis and mining 8, no. 1 (2018): 28., 2018.

Chen Sanghrajka, Hidalgo and Kapadia. Lisa: Lexically intelligent story assistant. proceedings of the 13th artificial intelligence and interactive digital entertainment conference, 2017.

Karin Kipper Schuler. Verbnet: A broad-coverage, comprehensive verb lexicon., 2005.

SemEval. Semeval portal (n.d.).

Spacy. spacy, 2016.

Staiano and M. Guerini. Depechemood: A lexicon for emotion analysis from crowd-annotated news, 2014.

Google SyntaxNet. Announcing syntaxnet: The world's most accurate parser goes open source, 2016.

Feride Savaroğlu Tabak and Vesile Evrim. Comparison of emotion lexicons., 2016.

Bhattacharyya P Tripathi V, Joshi A. Emotion analysis from text: a survey.

Jichen Zhu Valls-Vargas, Josep and Santiago Ontanon. Error analysis in an automated narrative information extraction pipeline.., 2017.

Santiago Ontanón Valls-Vargas, Josep and Jichen Zhu. Toward automatic character identification in unannotated narrative text., 2014.

K. Vonnegut. Shapes of stories., 1995.

Zaiane Yadollahi, Shahraki. Current state of text sentiment analysis from opinion to emotion mining. acm comput surv (csur) 50(2):25:1–25:33 ., 2017.

Kevin Hsin-Yih Lin Yang, Changhua and Hsin-Hsi Chen. Writer meets reader: Emotion analysis of social media from both the writer's and reader's perspectives., 2009.

Rui-Feng Xu Ye, Lu and Jun Xu. Emotion prediction of news articles from reader's perspective based on multi-label classification., 2012.

Rui-Feng Xu Ye, Lu and Jun Xu. Reader's emotion prediction based on weighted latent dirichlet allocation and multi-label k-nearest neighbor model., 2013.