ENRICHING AFFECT ANALYSIS THROUGH EMOTION AND SARCASM DETECTION

AMEETA AGRAWAL

A DISSERTATION SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

March 2018

Abstract

Affect detection from text is the task of detecting affective states such as sentiment, mood and emotions from natural language text including news comments, product reviews, discussion posts, tweets and so on. Broadly speaking, affect detection includes the related tasks of sentiment analysis, emotion detection and sarcasm detection, amongst others.

In this dissertation, we seek to enrich textual affect analysis from two perspectives: emotion and sarcasm. Emotion detection entails classifying the text into fine-grained categories of emotions such as *happiness*, *sadness*, *surprise*, and so on, whereas sarcasm detection seeks to identify the presence or absence of sarcasm in text. The task of emotion detection is particularly challenging due to limited number of resources and as it involves a greater number of categories of emotions in which to undertake classification, with no fixed number or types of emotions. Similarly, the recently proposed task of sarcasm detection is complicated due to the inherent sophisticated nature of sarcasm, where one typically says or writes the opposite of what they mean.

This dissertation consists of five contributions. First, we address word-emotion association, a fundamental building block of most, if not all, emotion detection systems. Current approaches to emotion detection rely on a handful of manually annotated resources such as lexicons and datasets for deriving word-emotion association. Instead, we propose novel models for augmenting word-emotion association to support unsupervised learning which does not require labeled training data and can be extended to flexible taxonomies of emotions.

Second, we study the problem of affective word representations, where affectively

similar words are projected into neighboring regions of an n-dimensional embedding space. While existing techniques usually consider the lexical semantics and syntax of co-occurring words, thus rating emotionally dissimilar words occurring in similar contexts as highly similar, we integrate a rich spectrum of emotions into representation learning in order to cluster emotionally similar words closer, and emotionally dissimilar words farther from each other. The generated emotion-enriched word representations are found to be better at capturing relevant features useful for sentence-level emotion classification and emotion similarity tasks.

Third, we investigate the problem of computational sarcasm detection. Generally, sarcasm detection is treated as a linguistic and lexical phenomena with limited emphasis on the emotional aspects of sarcasm. In order to address this gap, we propose novel models of enriching sarcasm detection by incorporating affective knowledge. In particular, document-level features obtained from affective word representations are utilized in designing classification systems. Through extensive evaluation on six datasets from three diverse domains of text, we demonstrate the potential of exploiting automatically induced features without the need for considerable manual feature engineering.

Motivated by the importance of affective knowledge in detecting sarcasm, the fourth contribution of this thesis seeks to dig deeper and study the role of transitions and relationships between different emotions in order to discover which emotions serve as more informative and discriminative features for distinguishing sarcastic utterances in text.

Lastly, we show the usefulness of our proposed affective models by applying them in a non-affective framework of predicting the helpfulness of online reviews.

To my beloved grandma and my precious baby girl,

and all the inspiring women in my life.

In memory of Prof. Nick Cercone.

"*Prediction is very difficult, especially about the future.*"

– Danish proverb

Acknowledgements

My PhD journey was made possible and enjoyable thanks to many wonderful people. The first person I would like to express my gratitude toward is my advisor, Prof. Aijun An. I sincerely thank you for showing faith in me all those years ago, and for continuing to inspire, encourage and support me over the years. Thank you for being the wind beneath my wings.

I would also like to thank my committee members, Parke Godfrey, Manos Papagelis, Xiaohui Yu, Steven Wang and Diana Inkpen. Your many insightful comments and interesting discussions not only helped me refine my work, but also provided exciting fodder for future research.

Inspired discussions, random walks and endless chats with my labmates have been crucial in preserving my ~~sanity~~ enthusiasm and sparking new ideas. I also owe you guys for continuing to make me feel a part of our lab even after my geographical displacement. A special shout-out to our graduate program staff and the rest of the technical team at CSE for their finest support.

Lastly, the endless love and support of my awesome friends and insanely cute family cannot be appreciated enough in words. To my dearest parents – thank you for always holding my hand while I tried to chase my dreams. To my sweetest in-laws – I became stronger when you joined me along for the journey. To *bhai* – thanks for being my cheerleader forever. To my little *chinchipie* – you motivated me in ways that you will never know. And above all, to my *soulmate* – we both know that I would have never made it without you, *arigatou gozaimasu*!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, there has been a great deal of interest in affect analysis which consists of methods for automatically identifying opinions, emotions, and sentiments in text. This wave of activity is due to the rapid growth of social media - product reviews, discussion forum posts, blogs, microblogs, and social networks - and the ensuing easy access to a mass of subjective and emotional data recorded in the digital format.

What other people think and how they feel plays a huge part in our decision making. "*What do you think about the new iPhone?*" or "*How did you enjoy this movie?*" or "*Could you recommend a good restaurant for brunch?*", are just a few types of questions one would encounter in everyday communication. Earlier, people would get this information through word-of-mouth from family and friends. But now, vast amount of opinions can be easily obtained from forums, blogs and review sites. As useful as this user-generated content is, finding, analyzing and summarizing all this data manually to get to the crux of the matter, however, is not a trivial task. Automatic affect analysis can lend a hand to improved decision-making.

In this dissertation, we study the field of affect analysis from two specific perspectives: **emotion recognition** and **sarcasm detection**. We begin this chapter by briefly introducing the task of affect detection from text along the axes of emotion and sarcasm in section 1.1. Next, section 1.2 describes the motivations and objectives of our work. Then, in section 1.3, we discuss briefly our research contributions. Finally, section 1.4 presents the outline of this thesis.

1.1 Affect Detection from Text

Broadly speaking, affect detection includes the related tasks of sentiment analysis, emotion detection and sarcasm detection, amongst others. While research in the field of sentiment analysis (typically consisting of binary classification into categories of *positive* and *negative* sentiments, and sometimes *neutral* as well) has received extensive attention over the last two decades, the emerging fields of multi-class emotion recognition (which entails classifying text into one or more categories of emotion such as *happiness*, *sadness*, *anger*, and many more) and sarcasm detection (where text is classified as being sarcastic or not), have remained largely underexplored due to numerous reasons elaborated in the following sections.

1.1.1 Emotion Detection

Emotion analysis provides a more challenging problem than the binary sentiment classification problem (Agrawal and An, 2013, 2014). While both tasks suffer from the subtleties that the implicit nature of language holds, emotion analysis is further complicated due to the greater number of categories (emotions) involved in which to

undertake classification. Additionally, there is no fixed number or types of emotions, as varying theories of emotion have been proposed by psychologists over the years, each detailing a slightly different set of emotions. Categorization into distinct emotion classes is more difficult not only because emotion recognition in general requires deeper insights, but also because of the innate similarities between different emotions which make clean classification a challenge. Particularly notable in this regard are *anger* and *disgust*, two emotion classes which even human annotators often find hard to distinguish (Aman and Szpakowicz, 2007).

Emotions from text can be recognized at various levels of granularity such as a word, a phrase, a sentence, a paragraph or even an entire document. A system designed to analyze emotions from a phrase, a sentence or a document is generally initialized with some form of word-level analysis, which is then scaled up to larger units of text. Consider an example sentence, "*Partying all weekend was such fun!*", consisting of emotion evoking words such as "*partying*" and "*fun*". No matter which technique (supervised machine learning, corpus-based, rules-based) is employed for detecting the emotion in the given sentence, an indispensable component of all the approaches is the degree of word-emotion association of the emotion evoking words present within the sentence. For instance, the words "*partying*" and "*fun*" are mostly associated with the emotion *happiness*. Although a word may evoke different emotions which can only be disambiguated in context (e.g., "funny movie" versus "funny tasting pasta"), most words can be found largely associated with a single emotion. Such emotion connotation is normally obtained from a precompiled emotion lexicon or derived automatically from large unannotated corpora of text which implicitly embed such associations.

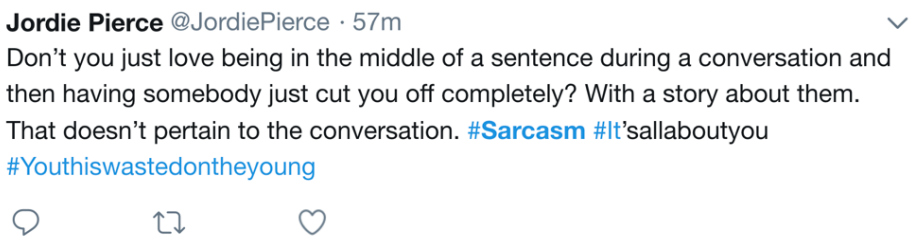


Figure 1.1: An illustrative tweet expressing sarcasm.

In this thesis, we focus on deriving and analyzing the emotion connotation of words, which forms the most basic yet crucial building block of most, if not all, emotion detection systems.

1.1.2 Sarcasm Detection

Sarcasm detection is a complicated task due to sarcasm being a sophisticated form of language where one typically says or writes the opposite of what they mean. The Merriam-Webster dictionary[1] defines sarcasm as a sharp and often satirical or ironic utterance designed to cut or give pain. Recently, Twitter has become a popular source of natural language text where people instantly broadcast their opinions and emotions to the world in 140 characters or less. Sarcasm, which is currently not an emotion category in any existing taxonomy, is generally exhibited quite freely in tweets. An illustrative tweet expressing sarcasm is provided in Figure 1.1.

Consider another tweet, *"Breakups are so much fun!"*. The fact that breakups are not fun at all is clearly comprehensible to humans, but for computers, such instances pose quite a challenge and are usually misclassified as they involve contrasting emotions in close proximity.

[1]https://www.merriam-webster.com/dictionary/sarcasm

In this thesis, we seek to address the challenging task of computational sarcasm detection. In particular, various algorithms incorporating affective information are proposed for detecting sarcasm in text.

1.1.3 Applications

The ability to detect emotions and sarcasm in text is critical for affect analysis systems which have numerous wide-ranging applications. One of the most popular applications of affect analysis is in market research. Analyzing sentiments or emotions expressed towards products allows companies to take the temperature of their customer base. For example, MotiveQuest taps into the big data of social media to get a feel for customer needs. Nike discovered that their customers want to feel accomplished, savvy, and powerful and enabled this with their "Just Do It" mentality, which has been a huge success. Another deep analysis showed that Dodge Ram customers wanted to feel powerful. So by concentrating on the truck's powerful features, the marketers were able to create a successful campaign that spoke to their customers' needs. The runaway success of Greek yogurt highlights another story where marketers have been able to connect with things that dieters and foodies care about - feeling accomplished, transformed, and pure.

Some other applications of affect analysis (Mohammad and Turney, 2013) include:

- Analyzing user-generated data: Automatically analyzing emotions hidden in innumerable comments on forums dedicated to different topics such as movies, consumer products, healthcare, financial services, politics, and numerous other entities, can help identify and understand the thoughts and feelings of peo-

ple. This knowledge can help interested parties to assess movie trends, address consumer complaints, provide better healthcare information, avoid stock market volatility, predict elections and so on. In short, understanding not only how people think, but also how they *feel* can assist in more informed decision-making (De Bondt et al., 2013; Desmet and Hoste, 2013; Dhall et al., 2013; Mahmud et al., 2017; Zheng et al., 2018).

- Customer relations: Automatic analysis of customer satisfaction can allow managers to take appropriate and timely actions (Gupta et al., 2013; Mohammad and Turney, 2013).

- Tutoring systems: It is believed that including affect recognition into e-learning platforms will help create more effective tutoring systems (Sun et al., 2013; Wiggins et al., 2014).

- Text-to-speech synthesis: Incorporating an emotional tone can facilitate more natural text-to-speech systems (Trilla and Alias, 2013).

- Human computer interaction: Assistive technology that is more sensitive to human emotions may not only be more useful, but also more acceptable (Hartmann et al., 2013; Siegert et al., 2013).

- Persuasive communication: By being able to convey a desired emotion, advertisers can create a powerful marketing tool (Shrum et al., 2013).

- Creative pursuits: Analyzing the emotional pulse of music, literature, art, and many more, could give a small insight into what makes humans tick (Huang and Lin, 2013).

In addition, correctly identifying sarcastic instances is useful in sorting out the intended affect expressed in text. For example, consider a product review, "*I love my Kindle - I have to leave my house and drive 10 minutes up the road to get it to work*". Although the review contains a strongly positive affective word, "*love*", the author is clearly attempting to convey anything but a positive tone. Accurately distinguishing sarcasm from literal meaning can strengthen affect analysis systems. Some other interesting applications of sarcasm detection include identifying sarcasm from student feedback collected via Twitter (Altrabsheh et al., 2015), indicating the progression of Huntington disease from social–cognitive tests (Larsen et al., 2016), and many more.

1.2 Motivations

In this section, we present the motivations behind our work while discussing the open issues in the emerging fields of multi-class emotion recognition and sarcasm detection.

1.2.1 Emotion Detection

Textual emotion detection is generally initialized with some degree of emotion connotation of words which can be defined as the type of emotion a word generally evokes or is associated with. Such association is normally derived from precompiled emotion annotated resources (such as labeled datasets or lexicons) or computed through statistical measures of semantic similarity or text representations from large unannotated corpora of text which implicitly embed such associations.

All document-level emotion detection systems, whether supervised or unsuper-

vised, harness emotion connotation of words, also called word-level emotion association, in some form or another. For instance, supervised models obtain this association directly during the classification model building phase from the labeled set of training data where each piece of text (e.g., sentence) is labeled with an emotion, or benefit from additional features in the form of emotion connotation extracted from lexicons or large corpora based on measures of semantic relatedness. Alternatively, unsupervised models not leveraging pre-classified training datasets, gather the word-emotion association through emotion lexicons or large unannotated corpora using measures of semantic similarity or text representations.

In essence, the emotion connotation of words forms the fundamental building block of most emotion detection systems, and is typically acquired from either or both of the following: (i) annotated resources such as labeled datasets or lexicons, or (ii) computed automatically through statistical corpus-based models of word associations or text representations from a large unlabeled corpus.

The problem with most existing manually constructed resources, namely datasets and lexicons, available for emotion detection, is their limited size mainly due to the expensive process of manual annotation. For example, two popular emotion datasets created by Alm (2008) and Aman and Szpakowicz (2007) contain around 1200 sentences each and only about 5000 unique words each. This limited coverage leads to the undesirable effect of leaving too many words unassociated with any emotion category. Another common source of word-emotion association is emotion lexicons, which contain lists of terms (i.e., words or phrases) and their degree of association with one or more emotions. However, current lexicons contain a small fixed number of emotions which makes them unsuitable for a larger taxonomy of

emotions (Du et al., 2014) or a newly defined different set of emotions (Facebook, 2016), and many words that bear emotions may not be in the lexicon.

At the same time, in order for emotion detection systems to be useful, they need to generalize well to diverse domains and applications by including a much larger vocabulary of words.

Thus, automatic statistical methods were proposed to derive the degree of association of a word with an emotion by leveraging large unannotated corpora of text. Such methods have the advantage of providing a wider vocabulary coverage of word-emotion associations, more flexibility in terms of emotion taxonomies, and requiring little to no human annotation. Automatic methods of computing emotion connotation can be roughly categorized as count-based methods such as Pointwise Mutual Information (PMI) (Church and Hanks, 1990) or word representation models such as `word2vec` (Mikolov et al., 2013a), both of which are based on co-occurrences of words in the training corpus. The former yields word-word and word-emotion associations or semantic relatedness scores between words and emotions, while the latter outputs representations of words using numerical values or word embeddings which can be used directly or to further compute word-word and word-emotion associations.

The count-based methods are largely unsupervised as they rely on co-occurrence counts and frequencies of words, and a handful of seed words that represent emotions. Word representation models, on the other hand, can be either unsupervised (based on the co-occurrences of words) or semi-supervised (using weakly labeled data).

However, there are several limitations of both count-based and word representation based models with respect to the task of emotion detection. First, the existing models drawing on the co-occurrence counts consider all the words occurring within a

context window of text equally, no matter how far they appear from one another. In addition, they do not discriminate between the left and right contexts of the words, thus failing to accommodate for any asymmetry in word associations. However, as noted by Tversky (1977), certain linguistic relationships are characteristically asymmetric, and we believe word-emotion association to be a type of such relationship. For instance, in one experiment to list the first meaningfully related word that comes to mind, for the cue word *fear*, 24% of the participants answered *scared*, while only 9% of them recalled *fear* when given the cue word *scared*, suggesting an inherent asymmetry in word-emotion associations (Altarriba et al., 1999). Lastly, by exclusively considering the co-occurrence contexts of words, such methods rate emotionally *dissimilar* words occurring in similar contexts such as "*happy*" and "*sad*" as more similar than the emotionally similar word pair "*happy*" and "*joy*", which is acutely problematic in affective tasks. We believe solving these problems can improve the accuracy of automatic methods for computing emotion connotation of words, and consequently, emotion detection systems at large.

1.2.2 Sarcasm Detection

Sarcasm detection from text has gained increasing attention over the last few years. However, if the problem of limited-sized labeled training data faced by emotion detection systems was considered challenging, then the problem of constructing manually labeled sarcasm datasets is even far more severe as sarcasm is a remarkably rare positive class (Abercrombie and Hovy, 2016), restricting the size of most datasets to a few hundred instances at most, which makes supervised learning with a simple

bag-of-words (e.g., n-grams) model an all but improbable course of action.

Therefore, additional features along with n-grams are employed. For instance, typical features used in sarcasm detection systems incorporate lexical elements such as word clusters, parts-of-speech, spelling and abbreviation, lengths of text, number of words, and so on.

However, although extensive research in psychology highlights a strong correlation between emotion and sarcasm (Filik et al., 2015; Larsen et al., 2016; Phillips et al., 2015; Riviello and Esposito, 2016), much of the existing work in computational sarcasm detection treats sarcasm as a lexical phenomena, with limited emphasis on the affective aspects of sarcasm. Therefore, in order to benefit from some level of affective knowledge, a few recent studies incorporated sentiment and emotion information into sarcasm detection models. For example, González-Ibáñez et al. (2011) introduced a sarcasm detection technique for tweets using numerous features derived from two affect lexicons and discovered a strong correlation between positive and negative emotions and sarcasm. Along the same lines, one study examining the distribution of emotions in sarcastic tweets found *trust* and *fear* to be the two most common emotions (Khokhlova et al., 2016), while a different study found *joy* and *anticipation* to be the two most common emotions (Sulis et al., 2016).

However, the few models leveraging affective information do so by relying upon a handful of existing annotated lexicons (sentiment or emotion), the limitations of which were outlined in the preceding subsection. Another major limitation of most of the existing approaches is their use of considerable manual feature engineering, a process that requires not only substantial time and effort, but also a reasonable level of domain expertise. Moreover, although one predominant notion concretizes sarcasm

as a manifestation of contrasting sentiments or emotions in close proximity, current systems fail to accommodate this property adequately. Lastly, and most importantly, despite the growing efforts in this emerging area, the biggest open issue which remains is that of low accuracy of the existing approaches. Thus, new methods are needed to address the aforementioned gaps and further improve the performance of sarcasm detection systems as a whole.

1.3 Research Contributions

This section briefly summarizes our core contributions. First, we propose two distinct models of automatically computing emotion connotation of words from large unannotated corpora of text, one supporting unsupervised learning, and the other supporting both supervised and unsupervised emotion detection and also benefiting from a semi-supervised paradigm. Then, we present two novel algorithms for computational sarcasm detection by specifically integrating affective knowledge and text representations, and also investigate the role of continuous transitions of affective content within sarcastic utterances. Lastly, we apply our affective models toward the design of a review helpfulness prediction system. In particular, the research contributions include:

- **Selective Co-occurrences for Word-emotion Association**:

  To solve the problem of count-based methods relying on the co-occurrences of words for deriving word-emotion association, we propose an unsupervised method, which we call Selective Co-occcurrences (SECO), motivated by certain

properties exhibited by emotions at large including mutual exclusivity, positional disposition and context weighting. The proposed approach, which can be initialized with as little as one seed word per emotion category, is found to be better at capturing the association between words and emotions than general purpose measures of semantic relatedness. Extensive evaluation of the word-emotion association scores derived from two large unannotated text corpora, Wikipedia articles and Amazon reviews, is conducted on three emotion datasets from very diverse domains. The proposed approach is particularly interesting as it requires no labeled training data and can be extended to flexible taxonomies of emotions.

- **Emotion-enriched Word Representations**:

  While SECO takes an unsupervised approach to learning word-emotion association requiring no manual input, our second contribution assumes a semi-supervised framework to further enhance models of word-level emotion connotation by integrating a small amount of affective knowledge derived from existing emotion lexicons. In order to address the foregoing issue of having emotionally dissimilar words estimated as more similar than emotionally similar words faced by current word representation models, the proposed method learns emotion-aware word representations, which we call Emotion Word Embeddings (EWE), by projecting emotionally similar words into neighboring regions of an n-dimensional embedding space. The proposed approach leverages distant supervision to automatically obtain a large training dataset of text documents with corresponding noisy emotion labels and two recurrent neural network ar-

chitectures for inducing the emotion-aware word representations. Extensive evaluation is carried out on two tasks including emotion classification and emotion similarity. In addition, a comparative study is presented measuring the performance of EWE against SECO.

- **Affective Word Embeddings for Sarcasm**:

  Our previous model, EWE, introduced affective text representations rendered along the axis of fine-grained emotion spectrum. In order to solve the problem of limited affective knowledge integrated into sarcasm detection systems and their heavy reliance on extensive manual feature engineering, our third contribution builds upon EWE by introducing a novel model, Affective Word Embeddings for Sarcasm (AWES), for automated sarcasm detection in text which encompasses not only the multi-category emotion taxonomies but also the binary polarities of sentiment models. The proposed methodology learns affective word representations from weakly labeled data, which are then employed for building sarcasm classifiers. Specifically, first, a large corpus of reviews is automatically labeled with noisy affective labels through distant supervision. Then, neural network models are trained to incorporate affective as well as contextual information into word representations. Extensive evaluation is conducted on six datasets across three domains of text (tweets, product reviews and discussion forum posts).

- **Emotion Transitions for Sarcasm**:

  In order to address the open issue of limited utilization of the property of contrasting affective states within close proximity as generally exhibited by sarcastic utterances, our next contribution seeks to fully exploit the importance of

affective knowledge in the task of sarcasm classification by formulating it as a sequence classification task. The proposed model, called Emotion Transitions for Sarcasm (ETS), leverages the transitions within multidimensional continuous emotional sequence data to detect sarcasm more efficiently. We experiment with a diverse set of emotions to discover which ones in particular are more distinctively effective at distinguishing sarcastic instances. We also evaluate our system on different forms of text such as tweets as well as product reviews to understand the inherent differences between sarcastic utterances in various genres. Lastly, a comparative study investigates the difference in performance between ETS and our previously proposed model AWES.

- **Application: Predicting Helpfulness of Online Reviews**:

  All of our preceding contributions engage in developing affective models evaluated in affective tasks such as emotion classification and sarcasm detection. In order to investigate the usefulness of our proposed affective models in a non-affective setting, we specifically apply them in the task of predicting review helpfulness.

  The advent of Web 2.0 has enabled users to share their opinions online easily, particularly in the form of online reviews. These reviews strongly influence the decision-making process. However, it is almost impossible to sift through all the available data in order to find the most relevant and helpful reviews. Predicting the helpfulness of reviews can help to save time and present helpful suggestions easily. Most previous works rely on extensive feature engineering involving factors such as user profile, semantic dictionaries and so on. The

role of an important factor, affective information, has received little attention. As our final contribution, we propose a novel method that leverages emotion information for building models for predicting the helpfulness of online reviews. The proposed approach is evaluated against several baselines on a large corpus of Amazon reviews data.

1.4 Thesis Outline

This thesis studies textual affect analysis from two perspectives: emotion and sarcasm. First, chapter 2 presents a brief background on neural networks and word representations, and chapter 3 surveys the recent literature in the fields of emotion recognition and sarcasm detection, highlighting the various resources and techniques generally employed.

Then, chapter 4 introduces an unsupervised statistical method of learning word-emotion association, called Selective Co-occurrences (SECO), from large unannotated text corpora by leveraging positional contexts and the property of mutual exclusivity. Through extensive evaluation, the effectiveness of the proposed approach is demonstrated over three emotion lexicons and two state-of-the-art algorithms of word embeddings on three datasets from different domains.

Chapter 5 presents a novel method of obtaining emotion-aware word representations, called Emotion Word Embeddings (EWE), which projects emotionally similar words into neighboring spaces. The proposed approach leverages distant supervision to obtain a large training dataset of text documents and two recurrent neural network architectures for learning the emotion-aware word representations. Extensive evalua-

tion on two tasks including emotion classification and emotion similarity demonstrates the effectiveness of the proposed approach over several competitive baselines.

Next, chapter 6 studies the problem of computational sarcasm detection by introducing a novel model called Affective Word Embeddings for Sarcasm (AWES). Extensive evaluation on six datasets across three domains of text (tweets, product reviews and discussion forum posts) demonstrates the effectiveness of the proposed model.

Chapter 7 presents another model for the task of sarcasm detection by formulating it as an emotion sequence classification problem. To leverage a richer set of features, the proposed model, Emotion Transitions for Sarcasm (ETS), exploits the natural transitions in a spectrum of emotions over the course of the text instance for distinguishing sarcastic utterances. Experiments conducted on two evaluation datasets demonstrate the potential of employing emotion transitions for sarcasm detection.

Furthermore, in chapter 8, we describe an application employing our proposed affective word representations in a non-affective framework by designing a model for predicting the helpfulness of online reviews.

Lastly, chapter 9 concludes the thesis, summarizing our main contributions, and describing directions for further research.

Chapter 2

Background

In this chapter, we provide a brief background description on the topics of neural networks and word representations. The content presented in this chapter is aimed at providing relevant context to the discussion in the following chapters.

2.1 Neural Networks

Neural networks have a long history. In what follows, we discuss some commonly used neural network architectures for learning word embeddings.

2.1.1 Feedforward Neural Networks

A feedforward neural network consists of an input layer, an output layer, and one or more hidden layers, with each non-input layer consisting of nodes or neurons - the essential basic processing unit of a neural network. Every unit in a layer is connected to all the units in the previous layer. However, each connection may have a different

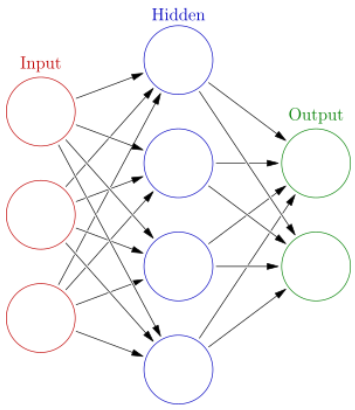


Figure 2.1: An example of a feedforward neural network

weight, and it is these weights on the connections that primarily encode the knowledge of a network. An example of a simple feedforward network with one hidden layer is depicted in Figure 2.1.

The first layer of a neural network, i.e., input layer, is used to provide the input data or features to the network, whereas the last layer, i.e., output layer, gives out the predictions.

A neuron calculates the weighted sum of its inputs and then applies an activation function to normalize the sum. The weights associated with each input of a neuron are the parameters which the network has to learn. During the learning phase, the training samples are passed through the network and the output obtained from the network is compared with the actual output. The differences between the actual outputs and the predicted outputs are propagated back through the layers to modify the connection weights. In other words, the backpropagation algorithm updates the weights of the neurons such that the error decreases gradually.

An activation function provides a normalizing effect on the output of the neuron by

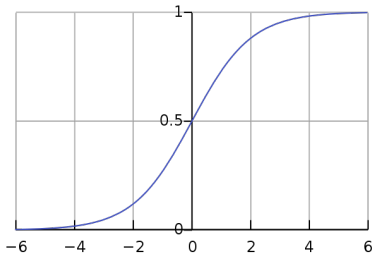


Figure 2.2: Sigmoid function.

preventing it from becoming very large after several layers. There are many activation functions such as sigmoid, tanh, Recitified Linear Unit (ReLU), etc. Specifically, for a binary classification problem such as sarcasm detection, the output is either 0 or 1. Thus, a sigmoid activation function is used. For a multi-class classification problem involving more than two classes, such as emotion classification, a softmax activation, which is a generalization of the sigmoid function to multiple classes, is used. Figure 2.2 shows the plot of a sigmoid function. Formally, the sigmoid function is computed as:

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \tag{2.1}$$

Feedforward neural networks can be viewed as a directed acyclic graph, without any feedback connections or loops in the network. They are ideally suitable for modeling relationships between a set of input variables and one or more output variables. In other words, a feedforward neural network is a non-sequential model, and are not ideal for classifying data which cannot naturally be presented as a vector of fixed dimensionality. That is why, a sliding window approach is typically used when applying feedforward neural networks to textual data.

However, in text processing the primary input variable is usually a document - an

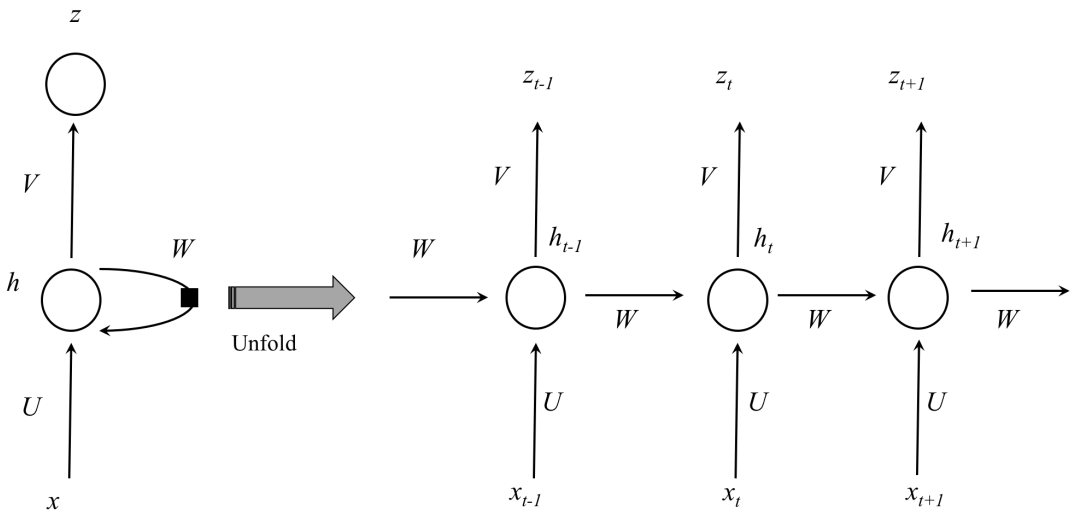


Figure 2.3: A recurrent neural network and the unfolding in time of the computation.

ordered sequence of words, of arbitrary length.

2.1.2 Recurrent Neural Networks

In a feedforward neural network, all the inputs are assumed to be independent of each other, without any notion of order. However, text documents exhibit sequential structure at different levels of abstraction such as paragraphs, sentences, phrases, words and so on, and it is useful to preserve the order of the data.

Unlike feedforward neural networks, the connections between units in a recurrent neural network (RNN) form a directed cycle. In other words, in each neuron of RNN, the output of the previous time step is fed as input to the next time step, allowing RNNs to process arbitrary sequences of inputs recursively. There is information in the sequence itself, and RNNs use it to perform tasks that feedforward networks cannot. Figure 2.3 depicts an overview of a recurrent neural network.

Traditional RNNs model sequential dynamics by mapping input sequences to hidden states, and hidden states to outputs. More formally, given an input sequence $x = \{x_1, x_2, ..., x_T\}$, a conventional RNN updates the hidden vector sequence $h = \{h_1, h_2, ..., h_T\}$ and output vector sequence $z = \{z_1, z_2, ..., z_T\}$ from $t = 1$ to T via the following recurrence equations:

$$h_t = \mathcal{H}\left(Wx_t + Uh_{t-1} + b_h\right) \tag{2.2}$$

$$z_t = Vh_t + b_z \tag{2.3}$$

where x_t is the input, h_t is the hidden state, z_t is the output at time t, U, V and W are the weight matrices, b is the bias term and $\mathcal{H}(\cdot)$ is the recurrent hidden layer function.

In theory, RNNs can make use of information in arbitrarily long sequences. However, in practice, it can be difficult to train simple RNNs to learn long-term dynamics as the components of the gradient vector can grow or decay exponentially over long sequences (Hochreiter and Schmidhuber, 1997).

2.1.3 Long Short-Term Memory Networks

To address the issue of *exploding* or *vanishing* gradients as faced by vanilla RNNs, Long Short-Term Memory networks (LSTM) were proposed, which explicitly allow the network to learn when to forget or update hidden states (Hochreiter and Schmidhuber, 1997). LSTM is a recurrent neural network that can model long-range word dependencies and handle inputs of varying lengths.

The LSTM model introduces a structure called a memory cell (see Figure 2.4), which is composed of four main elements: an input gate, a forget gate, an output

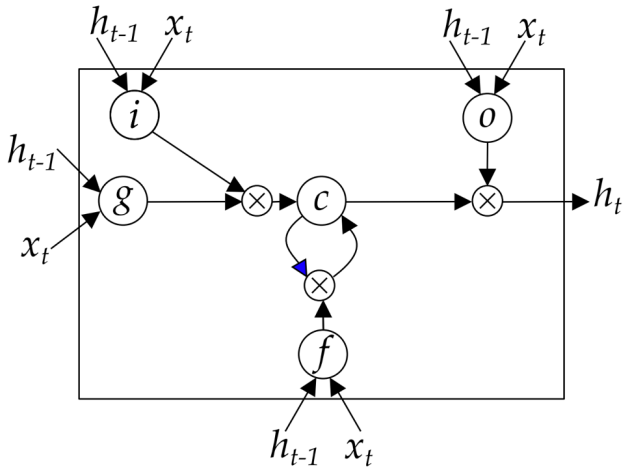


Figure 2.4: LSTM memory cell. i: input gate, f: forget gate, o: output gate, g: input modulation gate, c: memory cell. Adapted from (Sønderby and Winther, 2014).

gate and a cell state vector. The cell state vector ensures that, barring any outside interference, the state of a memory cell can remain constant from one timestep to another. The gates serve to modulate the interactions: the input gate can allow incoming signal to alter the state of the memory cell or block it; the output gate can allow the state of the memory cell to influence other neurons or prevent it; and, the forget gate allows the cell to remember or forget its previous state.

Since the original LSTM model, many variations have been proposed (Gers et al., 2000; Graves, 2013). One such variation (Theano Development Team, 2016), where the activation of a cell's output gate does not depend on the memory cell's state in order to facilitate faster training is described below. Given input x_t, the updates of

the LSTM cell at time step t are described by the following set of equations:

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{2.4}$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{2.5}$$

$$g_t = \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{2.6}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{2.7}$$

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{2.8}$$

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{2.9}$$

where $\odot$ denotes element-wise product, σ is the logistic sigmoid function, i_t, f_t and o_t are the input gate, forget gate and output gate, respectively, c_t is the cell state vector, W and U are the model's weight matrices and b is the bias term. Notably, the weights are reused at every time step, thus preventing the parameter size from growing proportionally to the sequence length. Additionally, due to the sigmoid non-linearity, which squashes the output to a $[0, 1]$ range, LSTM learns to selectively consider an input (i_t) or forget previous memory (f_t) or transfer memory to hidden state (o_t).

2.2 Word Representations

A word representation is a mathematical object associated with each word, often a vector (Turian et al., 2010). Traditional supervised bag-of-words model treat words as discrete atomic symbols, which are then transformed into a feature vector using a one-hot representation, where only one dimension's value is 1, with all the other being 0, and the feature vector has the same length as the size of the vocabulary. For instance, the word 'cat' may be represented as (0,0,0,0,1,0,0,...), and 'dog' may

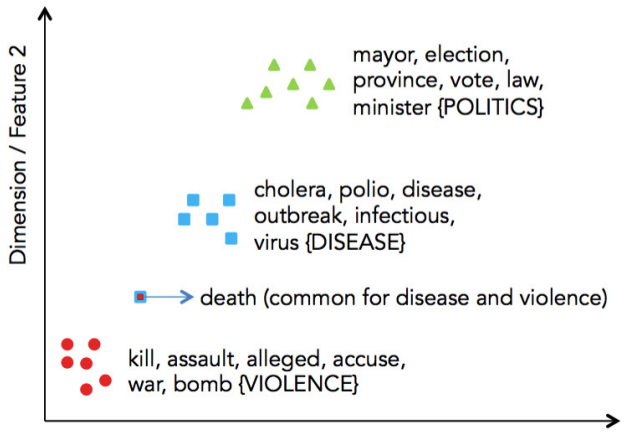


Figure 2.5: 2d visualization of sample embeddings

be (0,1,0,0,0,0,0,...). Such arbitrary encodings not only provide no useful information regarding the relationships that may exist between two words, but also lead to data sparsity.

One approach to overcoming some of the limitations of one-hot word representations is to use unsupervised methods for inducing word representations over large unlabeled corpora using neural probabilistic language models (Collobert et al., 2011; Turian et al., 2010). Such word representations, also known as word embeddings, are distributed representation of words where each word w in the vocabulary $\mathcal{V}$ is mapped into a dense, low-dimensional, continuous-valued vector $\mathbf{x} \in \mathbb{R}^n$ of dimensionality $n \ll |\mathcal{V}|$. Commonly, these word embeddings are obtained from the weight matrices of a neural network.

For a word said to be encoded as a 300 dimensional vector, there are 300 numbers used to describe its location in the continuous multi-dimensional embedding space. For instance, 'cat' may be represented as (0.50, 0.22, 0.73, 0.11, ...) and 'dog' may be (0.32, 0.83, 0.44, 0.29, ...). Figure 2.5 shows an example 2d visualization of an

embedding space (Agrawal et al., 2016).

Word embedding models follow the distributional hypothesis, which states that words appearing in similar contexts tend to be related. In other words, semantically similar words are mapped to, or embedded in, neighboring regions of the vector space.

2.3 Learning Word Representations

In this section, we present an overview of the skip-gram neural network model, popularized by the `word2vec`[1] program, for learning word representations from large text corpus.

The skip-gram model uses a feedforward neural network with a single hidden layer during the training phase, and the weights of the hidden layer denote the word representations or word embeddings. In other words, the end goal really is to just learn the hidden layer weight matrix.

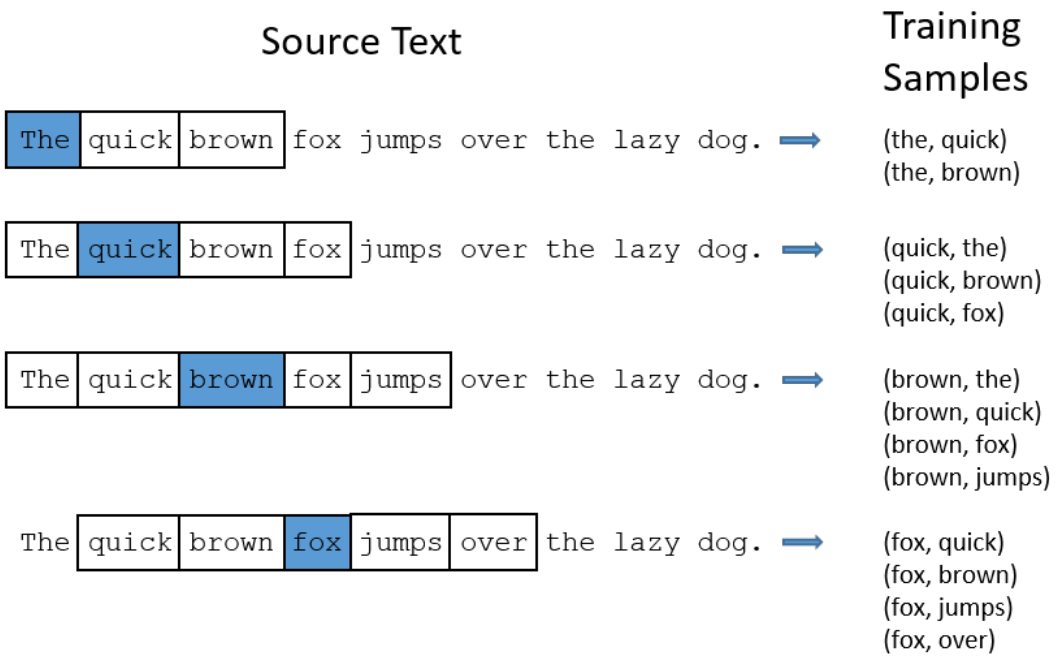
Basically, given a very large set of training sentences or documents (i.e., a corpus), the skip-gram model tries to predict the neighboring words given a target word within a context window of text of fixed size $2k + 1$, centered around the target word. As an illustration, consider the following sentence:

The quick brown fox jumps over the lazy dog

Assuming $k = 2$ (i.e., at most 2 context words on either side of the target word), the following windows of text and the corresponding training samples (word pairs) can be generated[2]:

[1] https://code.google.com/archive/p/word2vec/

[2] https://towardsdatascience.com/word2vec-skip-gram-model-part-1-intuition-78614e4d6e0b



For each word pair (input, output), the model predicts the probability for every word in the vocabulary of being the output context word for a given input target word. The output probabilities indicate how likely it is to find each vocabulary word in the context of the input word. For example, if you gave the trained network the input word 'car', the output probabilities are going to be much higher for words like 'vehicle' and 'engine' than for unrelated words like 'chocolate' and 'spider'.

Note that although skip-gram uses a neural network model during the training phase, it is still considered to be an unsupervised model as it generates the input and output variables from a piece of text, without requiring any manual expertise in the form of labeled data.

As an example, assume $\mathcal{V} = \{w_1, w_2, ...w_V\}$ to be the vocabulary of the entire corpus, and $V = 10000$ denotes the size of the vocabulary $\mathcal{V}$. Consider the input to the neural network to be the word 'ants' represented as a one-hot vector of dimen-

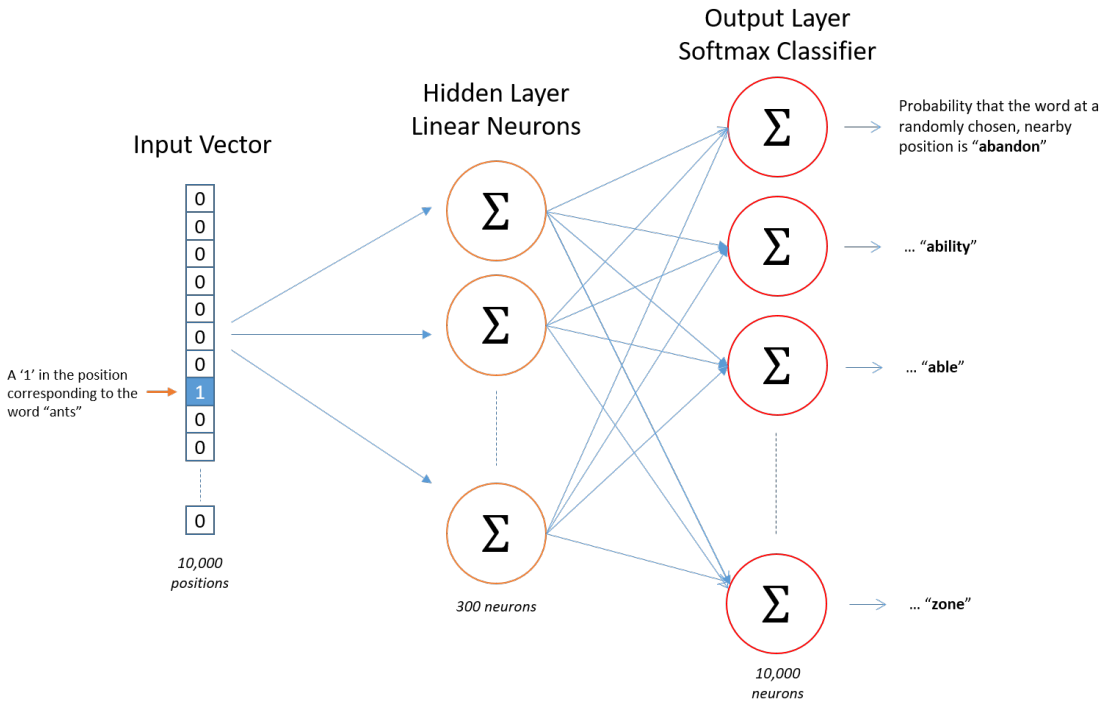


Figure 2.6: Skip-gram's neural network architecture.

sion 10,000. The output of the network will be a single vector of dimension 10,000 containing, for every word $w_i \in \mathcal{V}$, the probability that a randomly selected word is a neighboring word. An overview of the architecture of the skip-gram's neural network is presented in Figure 2.6[3].

Let's say the learned word vectors are of length (i.e., dimensions or features) 300[4]. Therefore, the hidden layer will be represented by a weight matrix with 10,000 rows (i.e., V) and 300 columns (one for every hidden neuron). Finally, the 1×300 word vector for 'ants' is fed to the output layer, which is a softmax regression classifier.

In a nutshell, and as we will observe later in this thesis, although the underlying

[3]https://towardsdatascience.com/word2vec-skip-gram-model-part-1-intuition-78614e4d6e0b

[4]The pre-trained skip-gram vectors obtained from the Google news dataset contain 300 features. This is a hyper-parameter, typically in the range of 50 to 500.

architecture for training (i.e., feedforward vs. recurrent neural networks, number of hidden layers, choice of activation function, etc.) may differ for different word embedding models, the object of importance – the learned word embeddings, is still obtained from the weight matrix of the hidden layer of a neural network.

Chapter 3

Literature Review

In this chapter, we survey the state-of-the-art research in the area of affect analysis for text documents along two axes: emotion and sarcasm.

3.1 Emotion Detection

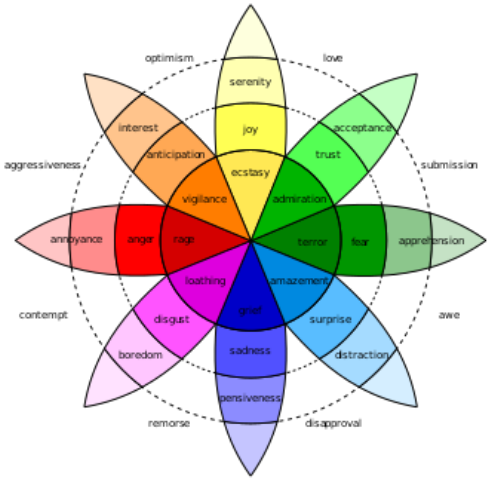
We begin this section by briefly presenting the prevailing theories of emotion from a psychological point of view. Then, we discuss the resources and techniques previously employed for document-level emotion classification and word-level emotion association.

3.1.1 Theories of Emotion

Although there is no strict definition for emotion, most researchers agree that it is a particular feeling that characterizes a state of mind, such as *joy*, *anger*, *fear* and so on. Emotion analysis from text is the task of identifying emotions from natural

Primary emotion	Secondary emotion	Tertiary emotions
Love	Affection	Adoration, affection, love, fondness, liking, attraction, caring, tenderness, compassion, sentimentality
	Lust	Arousal, desire, lust, passion, infatuation
	Longing	Longing
Joy	Cheerfulness	Amusement, bliss, cheerfulness, gaiety, glee, jolliness, joviality, joy, delight, enjoyment, gladness, happiness, jubilation, elation, satisfaction, ecstasy, euphoria
	Zest	Enthusiasm, zeal, zest, excitement, thrill, exhilaration
	Contentment	Contentment, pleasure
	Pride	Pride, triumph
	Optimism	Eagerness, hope, optimism
	Enthrallment	Enthrallment, rapture
	Relief	Relief
Surprise	Surprise	Amazement, surprise, astonishment
Anger	Irritation	Aggravation, irritation, agitation, annoyance, grouchiness, grumpiness
	Exasperation	Exasperation, frustration
	Rage	Anger, rage, outrage, fury, wrath, hostility, ferocity, bitterness, hate, loathing, scorn, spite, vengefulness, dislike, resentment
	Disgust	Disgust, revulsion, contempt
	Envy	Envy, jealousy
	Torment	Torment
Sadness	Suffering	Agony, suffering, hurt, anguish
	Sadness	Depression, despair, hopelessness, gloom, glumness, sadness, unhappiness, grief, sorrow, woe, misery, melancholy
	Disappointment	Dismay, disappointment, displeasure
	Shame	Guilt, shame, regret, remorse
	Neglect	Alienation, isolation, neglect, loneliness, rejection, homesickness, defeat, dejection, insecurity, embarrassment, humiliation, insult
	Sympathy	Pity, sympathy
Fear	Horror	Alarm, shock, fear, fright, horror, terror, panic, hysteria, mortification
	Nervousness	Anxiety, nervousness, tenseness, uneasiness, apprehension, worry, distress, dread

(a)



(b)

Figure 3.1: Taxonomies of emotions: (a) Parrott's, (b) Plutchik's.

language data such as news articles, blog posts, user reviews, and many more.

Over the years, psychologists have proposed a number of theories that classify human emotions into taxonomies. Primarily, there are two popular theoretical approaches in emotion research. The categorical approach defines emotional states by discrete classes, whereas the dimensional approach locates different emotions in a two- or three-dimensional space, with the most often used dimensions being activation (arousal), potency (power), and evaluation (pleasure). In this thesis, we focus on the categorical models of emotions, which are well-represented in the field of computational emotion detection. Figure 3.1 presents two categorical models of emotions.

Izard (1971) defined an emotion taxonomy consisting of ten basic emotions, namely *anger*, *contempt*, *disgust*, *distress*, *fear*, *guilt*, *interest*, *joy*, *shame* and *surprise*. Then, Plutchik (1980) proposed a theory consisting of eight emotions including *joy*, *sadness*,

anger, *fear*, *disgust*, *surprise*, *trust* and *anticipation*. In computational linguistics, one of the most widely used model is the one proposed by Ekman (1992), which is, as a matter of fact, a subset of Plutchik's list, including six basic emotions: *joy*, *sadness*, *anger*, *fear*, *disgust*, and *surprise*. A more recent theory (Parrott, 2001) also put forth a set of six emotions such as *love*, *joy*, *surprise*, *anger*, *sadness* and *fear*. Parrott's theory is in fact a slightly modified version of Ekman's theory, where the emotion *disgust* is replaced by the emotion *love*.

3.1.2 Document-level Emotion Classification

Much of the existing literature classifies techniques of emotion classification at document (or sentence) level into four main categories: keyword-based, rules-based, supervised and unsupervised (also known as corpus-based).

Keyword-based techniques predominantly leverage existing emotion lexicons and a simple keyword-matching algorithm to map the words in a document to their emotion categories, where the emotion with the maximum score is output as the final emotion label (Strapparava and Mihalcea, 2008). Rules-based techniques, although now less common, typically use hand-crafted rules along with emotion lexicons to capture the linguistic nuances (Chaumartin, 2007; Krcadinac et al., 2013; Neviarouskaya et al., 2009, 2010, 2011, 2013; Smith and Lee, 2013).

Supervised approaches apply machine learning algorithms such as Support Vector Machine (SVM), Naive Bayes (NB), etc., to labeled training datasets in order to automatically learn the relationships between features such as n-grams (e.g., unigrams, bigrams, trigrams), punctuation, etc. and the different classes of emotion categories

(Alm, 2008; Aman and Szpakowicz, 2008; Chaffar and Inkpen, 2011; Danisman and Alpkocak, 2008; Ghazi et al., 2010; Katz et al., 2007; Özbal and Pighin, 2013; Seol et al., 2008; Strapparava and Mihalcea, 2008).

Unsupervised approaches do not require any existing labeled resource such as training dataset. Instead, they adopt a statistical corpus-based technique making use of large unannotated text corpora such as Wikipedia or Gutenberg to induce word-emotion association automatically using some measure of semantic similarity such as Pointwise Mutual Information (PMI) (Church and Hanks, 1990), Latent Semantic Analysis (LSA) (Deerwester et al., 1990), to name a few.

A brief description of an unsupervised approach to document-level emotion detection is as follows. Let $d = \{w_1, w_2, ..., w_n\}$ denote a document or a sentence for which one wishes to obtain an emotion label. Assume $E = \{e_1, e_2, ..., e_m\}$ to be a finite discrete set of m emotion labels. Each emotion category $e_k \in E$ is represented by a set of t seed words[1] $s_k = \{s_{k1}, ..., s_{kt}\}$, where $1 \leq t$.

1. First, for every word, $w_i \in d$, and every seed word, $s_{kj} \in s_k$, representing an emotion category e_k, a word-word association, $\mathcal{A}(w_i, s_{kj})$, is computed through some metric of semantic relatedness. For example, $\mathcal{A}(\text{birthday}, \text{party}) = 0.43$ as obtained through LSA[2].

[1]Although these seed words are normally provided manually, the effort required is negligible compared to creating manually labeled training dataset. Thus, in general, such approaches are still considered as unsupervised.

[2]These LSA values were obtained from the LSA website: http://lsa.colorado.edu, with the semantic space (i.e., corpora) labeled as General Reading upto 1st year college used for the similarity computation.

2. Next, a word-emotion association $\mathcal{E}$ is calculated by averaging the word-word association scores for each word with respect to all the seed words of an emotion category. That is, $\mathcal{E}(w_i, e_k) = \frac{1}{t}\sum_{j \in t} \mathcal{A}(w_i, s_{kj})$. Essentially, each word is thus represented as a vector of m values, one for each emotion, i.e., $\phi(w_i) = \langle \mathcal{E}(w_i, e_1), \mathcal{E}(w_i, e_2), ..., \mathcal{E}(w_i, e_m) \rangle$.

3. Then, a document-level emotion vector is obtained by computing the average of the word-emotion association scores of all its words, i.e., $\phi(d) = \frac{1}{n}\sum_{i \in n} \phi(w_i)$.

4. Finally, the document is labeled with the emotion with the maximum score. In case of a tie, under single-labeling scheme, one possible solution includes selecting at random one label from the set of all labels with the highest value. Alternatively, if following a multi-labeling scheme, all the emotion labels with the maximum score can be output. Moreover, if the classification scheme enables the *neutral* category, then the document is considered neutral if no emotion category's score is greater than a certain specified threshold.

One of the earliest applications of PMI for calculating semantic association was proposed in the PMI-IR algorithm (Turney, 2001). Specifically in the task of emotion detection, PMI has been previously used to classify emotions in news headlines, where the probabilities of words were calculated using statistics collected from three search engines (MyWay, AlltheWeb and Yahoo) using queries composed of the sentence text and the emotion categories (Kozareva et al., 2007). Wikipedia has also proven a useful resource for calculating word frequencies using PMI to obtain word-emotion association scores (Agrawal and An, 2012). Alternatively, PMI has been used to first build an emotion lexicon which is then used for classification (Yang et al., 2007).

Their approach involved measuring collocation by multiplying the PMI of each word-emotion pair with the raw total count of co-occurrences of the word-emotion pair. The corpus comprised of over 1 million blog sentences, each containing one of 40 emoticons, with each emoticon representing an emotion category. LSA has also been used to calculate word-emotion association scores for classifying news headlines (Strapparava and Mihalcea, 2008).

Note that this categorization is more of a guideline than a strict classification as nowadays, most approaches employ a hybrid of some of these techniques in some form or another (Agrawal and An, 2012; Ghazi et al., 2014).

No matter which of the abovementioned techniques is employed for detecting the emotion in a given document, an indispensable component of all the approaches is the degree of word-emotion association, which is derived from annotated resources such as labeled datasets or emotion lexicons, or computed automatically through statistical approaches from a large unlabeled corpus of text.

3.1.3 Word-level Emotion Association

As noted in the preceding section, in essence, word-emotion association forms a fundamental building block of most, if not all, emotion detection systems. Next, we discuss the various ways through which word-emotion association is generally derived.

3.1.3.1 Manual Annotation

Annotation is the methodology used for creating resources (e.g., lexicons or datasets) by adding information such as the emotion label(s) at some level of text – a word, phrase, sentence or a document. Manual annotation is considered to be reliable,

although very labor-intensive.

Conventionally, more than one person annotates the same text to construct gold standard corpora, which are further used for creating, training and evaluating automatic models.

In order to determine how well two or more annotators agree on a given annotation, an inter-annotator agreement score is calculated. Given the extremely subjective nature of emotions and sarcasm, it is unsurprising, therefore, that the inter-annotator agreement of most emotion and sarcasm datasets falls in a modest range. For instance, the emotion blogs dataset created by Aman and Szpakowicz (2007) has an agreement in the range of 0.6 to 0.79; the sarcasm Amazon product reviews dataset created by Tsur et al. (2010) reached an agreement of 0.34; and, the agreement of sarcasm tweets dataset (Davidov et al., 2010) stands at 0.41.

To identify markers of emotions, many lexicons (i.e., lists of words, phrases, idioms) have been created. One of the earliest and most popular emotion resources is the WordNet Affect (Strapparava and Valitutti, 2004), developed by manually labeling about 1,314 synsets with one or more of Ekman (1992) six basic emotions. Fuzzy Affect Lexicon (Subasic and Huettner, 2001) contains 3,876 entries annotated manually with emotion labels, where the most frequent emotions are *conflict* and *violence*, and the least frequent include *health*, *sickness* and *facilitation*.

Created using crowd-sourcing, one of the largest manually annotated emotion lexicons is the NRC Emotion Lexicon (EmoLex) (Mohammad and Turney, 2010, 2013). It contains about 14,200 unigrams annotated with one or more of Plutchik's eight emotions (Plutchik, 2001). Another manually created lexicon, the Affect database (Neviarouskaya et al., 2007), contains a total of 2,440 entries including emoticons,

acronyms, words from WordNet Affect and modifiers annotated by three annotators using nine emotion labels and is the only manual lexicon to include the degree of word-emotion association scores in terms of real-valued intensities.

More recently, DepecheMood (Staiano and Guerini, 2014), was created using supervised training by applying distributional semantics to a dataset of crowd-annotated news articles. This lexicon consists of 37,000 words and their emotion scores across seven emotions.

Considering the fundamental role played by lexical resources in the task of emotion detection, the current options available in the form of emotion lexicons seem rather limited in their vocabulary coverage. Manual annotation, including crowd-sourcing, requires considerable lexicographic expertise, time and effort. Apart from containing a limited number of words, another limitation of manually created resources is that they also contain a fixed number of emotion categories, which makes them inapplicable for a different set of emotions (Du et al., 2014; Facebook, 2016).

Similarly, due to the subjective nature of emotions and the difficulties involved in creating manually labeled resources, the size of most currently available emotion annotated datasets is restricted to a few thousand instances. For that reason, supervised emotion detection models leveraging labeled datasets cannot generalize to a larger vocabulary.

3.1.3.2 Unsupervised Statistical Corpus-based Approaches

In order to overcome the drawbacks of manually annotated resources, an alternative approach involves unsupervised learning, where statistical corpus-based models leverage large unannotated corpora of text to automatically acquire word-emotion

association scores. Such techniques can address the problem of unseen vocabulary to a large extent, and as they employ a handful of emotion seed words at most to initialize the process, they are also easily applicable to flexible taxonomies of emotions. Moreover, they require little to no manual annotation, which is an expensive process.

All statistical corpus-based approaches are fundamentally based upon the intuitive assumption of distributional semantics, which states that co-occurring words in similar contexts tend to be related to each other. In other words, they follow the principle that a word's meaning can be induced by observing its statistical usage across a large sample of language.

Basically, unsupervised statistical approaches leverage co-occurrence statistics and context windows of text in order to automatically compute the relatedness between words. Many models of computing word semantic relatedness exist, from the traditional count-based methods such as Pointwise Mutual Information (PMI) (Church and Hanks, 1990) to the more recent neural-network inspired models of context-based word embeddings such as continuous bag-of-words (CBOW) (Mikolov et al., 2013a,b; Pennington et al., 2014).

The count-based methods leverage co-occurrence statistics and frequencies of words in large text corpora to compute their relatedness, whereas the neural network models project similar words into neighboring regions of some n-dimensional space to obtain continuous word representations, and then compute the association between words by using, for instance, a metric such as cosine similarity.

- **Count-based Methods**: Although most studies conclude that there is not one semantic relatedness measure that is best in all situations (Niraula et al.,

2015), Pointwise Mutual Information (Church and Hanks, 1990) and Latent Semantic Analysis (Deerwester et al., 1990) are traditionally popular in the area of emotion detection.

Despite, and perhaps because of, its simplicity, Pointwise Mutual Information (PMI) has long been a popular measure of semantic relatedness (Church and Hanks, 1990). It estimates the similarity between two terms x and y as:

$$\mathrm{PMI}\,(x, y) = \log \frac{p\,(x, y)}{p\,(x)\,p\,(y)} \tag{3.1}$$

where $p\,(x, y)$ is the probability that words x and y co-occur within a window of specific length, and $p\,(x)$ and $p\,(y)$ are the individual probabilities of word x and word y, respectively, in the corpus. The maximum value of this measure is determined by the minimum value between $-\log p\,(x)$ and $-\log p\,(y)$, and the minimum value, which happens when the number of co-occurrences of two words is zero, is $-\infty$.

To overcome a few well-known shortcomings of PMI (i.e., low frequency events receiving relatively high scores, lack of a fixed upper bound), Bouma (2009) proposed a normalized version of PMI, called Normalized PMI (NPMI), where:

$$\mathrm{NPMI}\,(x, y) = \frac{\mathrm{PMI}\,(x, y)}{-\log p\,(x, y)} \tag{3.2}$$

with fixed orientation values: when two words only occur together, $\mathrm{NPMI}(x, y) = 1$; when they are distributed as expected under independence, $\mathrm{NPMI}(x, y) = 0$; and, when they occur separately but not together, $\mathrm{NPMI}(x, y) = -1$.

Another measure of semantic analysis, Latent Semantic Analysis (LSA) (Deerwester et al., 1990), is based on a collection of different documents, where a doc-

ument could be, for instance, a paragraph in a large corpus. It represents words as vectors in a word-by-document space and analyzes the statistical relationships among words using Singular Value Decomposition (SVD) dimensionality reduction technique. First, a matrix $\mathbf{X}$, where the row vectors represent words, the column vectors represent documents, and the cells contain the weight (e.g., raw frequencies, *tf-idf* score) of the word, is constructed. Then, to reduce the high dimensional semantic space, SVD is applied to $\mathbf{X}$, thus decomposing it into a product of three matrices, which is further compressed to finally obtain matrix $\mathbf{Z}$ of rank k to best approximate the original matrix. Finally, the similarity between two words is estimated as the cosine of the angle between their corresponding compressed row vectors. Algorithm 1 summarizes the pseudocode of LSA.

Algorithm 1 Pseudocode of LSA (Budiu et al., 2007)

Input: a set of words W, a set of documents D and a number of factors k

1. Compute the matrix X of word-by-document occurrences: $X[i, j]$ represents how many times word i occurs in document j.
2. Compute LX from X such that $LX[i, j] = \log(1 + X[i, j])$.
3. Compute the entropy $H[i]$ of word i as: $H[i] = \sum_j -X[i, j] \log X[i, j]$.
4. Normalize the entries in LX as: $N[i, j] = LX[i, j]/H[i]$.
5. Use SVD on LX to obtain a matrix Z of dimensionality k.
6. A word i is represented as the vector $Z[i]$ and the similarity between words i and j is $\cos(Z[i], Z[j])$.

- **Contextual Word Representations**: More recently proposed neural-network

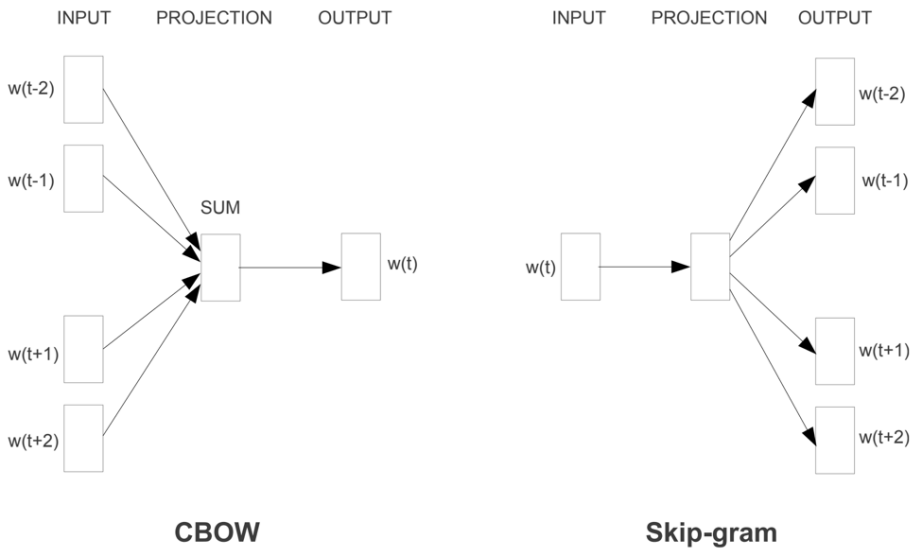


Figure 3.2: Model architectures of CBOW and skip-gram (Mikolov et al., 2013a).

based approaches, such as continuous bag-of-words (CBOW) and skip-gram (SG) (Mikolov et al., 2013a,b) output word representations which can then be used to compute the similarity between two words by calculating their cosine similarity. These methods implicitly factorize a word-context matrix whose cell values are in fact shifted PMI (Levy and Goldberg, 2014).

Contextual word representations, also referred to as generic word embeddings, are semantically meaningful floating point representations of terms learned from unannotated text. The models learn to represent each term as a fixed length embedding vector by predicting adjacent terms in the document (Bengio et al., 2003; Collobert and Weston, 2008; Collobert et al., 2011; Mikolov et al., 2013a,b; Pennington et al., 2014). The objective function drives the model to learn similar embedding vectors for semantically related words that appear in similar contexts.

Collobert et al. (2011) proposed a simple but effective feed-forward neutral network that independently classifies labels for each word by using contexts within a window with fixed size. Popularized by the `word2vec` toolkit, continuous bag-of-words (CBOW) and skip-gram (Mikolov et al., 2013a,b) use a neural network architecture for learning word embeddings from unsupervised data. The CBOW model predicts a target word (e.g., 'mat') given its surrounding context terms (e.g., 'the cat sits on the'), whereas the skip-gram model predicts the surrounding context words given a target word. Figure 3.2 depicts an overview of the CBOW and skip-gram model architectures.

Another word embedding model, Global Vectors (GloVe) (Pennington et al., 2014), estimates word representations by using them to construct a word-word co-occurrence matrix collected from a large text corpus with a weighting function on word pairs.

Note that, although the abovementioned models leverage neural networks for training the word embeddings, these are still considered as *unsupervised* approaches (Turian et al., 2010). That is because the word unsupervised in this context is with respect to how the training data is generated, i.e., based only on the co-occurrences and the surrounding contexts of words, without the need for any external source of manually created labeled data.

All the unsupervised contextual models such as PMI, LSA, CBOW, etc., assume the association between two words w_1 and w_2 to be symmetric, i.e., $Assoc.(w_1, w_2) = Assoc.(w_2, w_1)$. Certain linguistics relationships, however, are characteristically asymmetric (Tversky, 1977). For example, given a word such as *spinach*, one may recall

the word *green* more promptly than think of *spinach* given the cue word *green*.

Furthermore, these methods consider all the words occurring within a context window to be equally related, whereas in reality, words that appear nearer to each other have been found to exhibit stronger relationships (Beeferman et al., 1997).

In addition, by exclusively considering the co-occurrence contexts of words, both count-based and contextual word representations such as PMI and CBOW rate emotionally *dissimilar* words occurring in similar contexts such as "*happy*" and "*sad*" as more similar than the emotionally similar word pair "*happy*" and "*joy*", which is severely disadvantageous in affective tasks such as emotion classification.

As a result, while the existing unsupervised statistical models provide promising results on most word similarity tasks, they do not adapt well to the problem of emotion detection. In this thesis, therefore, we seek to address their limitations and propose a novel approach for computing word-emotion association scores. Motivated by certain properties exhibited by emotions such as that of mutual exclusivity (where a word is strongly associated with only one emotion in a given context) as well as context weighting (where nearer words are more strongly associated than words that are farther away), we tailor conventional measures of semantic relatedness for computing word-emotion association scores through the algorithm of selective co-occurrences (described later in chapter 4).

3.1.3.3 Weakly Supervised Statistical Corpus-based Approaches

Although the previously discussed unsupervised statistical models of learning word-emotion association overcome many disadvantages of manually annotated resources, they face their own set of limitations, one of which includes low accuracy. The per-

formance of unsupervised methods can be improved by incorporating a small amount of task-specific knowledge along with the contextual information. We describe the related work in the domain of task-specific affective word representations and the weakly labeled data used to obtain them below.

- **Task-specific Affective Word Representations**: To increase the effectiveness of generic word embeddings, there have been some lines of work in using neural networks for inducing task-specific *affective* embeddings. Socher et al. (2011) learned vector space representations for multi-word phrases using recursive autoencoders for the task of sentiment analysis. Labutov and Lipson (2013) produced task-specific embeddings from existing word embeddings for sentiment analysis. Kalchbrenner et al. (2014) trained their models on a large dataset of 1.6 million tweets, where a tweet was automatically labeled as positive or negative depending on the emoticon that occurs in it. Tang et al. (2016, 2014) also induced embeddings from scratch for sentiment analysis using a dataset of 10 million tweets obtained through distant supervision labeled with positive and negative emoticons. More recently, affective word representations have been obtained using a corpus of almost 1 billion tweets weakly labeled with a set of 64 emojis (Felbo et al., 2017).

  An alternative to learning task-specific embeddings from scratch or updating existing embeddings using neural networks is post-processing or fine-tuning the existing embeddings with respect to some external knowledge source such as a lexicon. For example, Faruqui et al. (2015) updated existing word vectors by using graph structures to propagate information derived from lexicons among

semantic concepts. External lexicons have also been exploited for word similarity and relational knowledge (Bian et al., 2014; Fried and Duh, 2014; Xu et al., 2014; Yu and Dredze, 2014) to improve `word2vec` embeddings in a joint training model.

- **Weakly Labeled Data**: Semi-supervised learning of word-emotion association requires weakly labeled data. Distant supervision has been shown to be an effective way of obtaining a large set of weakly labeled data (Go et al., 2009). Several studies have attempted to automatically gather very large training sets by exploiting reasonable indicators of emotional content (e.g., emoticons, hashtags, keywords) in text as noisy labels.

  Choudhury et al. (2012) automatically collected tweets labeled with mood hashtags appearing at the end of the tweet to analyze users' emotional states in social media. Distant supervision was also applied to collect tweets marked with hashtags (Mohammad, 2012) and emoticons along with hashtags (Purver and Battersby, 2012) corresponding to Ekman's six emotions. Wang et al. (2012) obtained a large emotion-labeled dataset of tweets by harnessing emotion-related hashtags for seven emotion categories, while Suttles and Ide (2013) collected tweets using emoticons, hashtags and emojis according to the set of eight emotions defined by Plutchik (1980).

  Mohammad and Kiritchenko (2015) collected tweets using hashtags for highly fine-grained emotions (585 emotion labels) to create a large emotion lexicon and found emotion-word hashtags to be suitable proxy labels of emotions in tweets. Most recently, Facebook reactions have also been used as proxies for emotion

labels for collecting posts from Facebook (Pool and Nissim, 2016).

All the above-mentioned approaches of learning task-specific affective embeddings (Felbo et al., 2017; Tang et al., 2016, 2014) rely on tweets data obtained from Twitter, automatically labeled using emoticons. However, tweets data do not generalize well to texts from other domains such as blogs, narratives, etc. Moreover, affective indicators such as emoticons previously leveraged for distant supervision are very domain-specific and do not occur in all types of data.

In this thesis, we seek to explore a novel domain of text (product reviews) to present a more generalizable approach to obtaining large-scale training data using distant supervision. In addition, while previous embeddings were trained on corpora of sizes ranging from 10 million to 1 billion tweets, our models are able to learn rich representations from a much smaller dataset of about 200K reviews. Furthermore, although a binary spectrum of positive and negative sentiment (Tang et al., 2014) or a large axis of 64 emojis (Felbo et al., 2017) has been previously used to generate representations, we align our embeddings along an emotion model firmly grounded in psychology which remains unexplored as yet. Lastly, while all the previous approaches used only a single-label setting (i.e., only one affect label per document), we propose modeling a more natural multi-label setting where a document can be associated with more than one emotion label (described further in chapters 5 and 6).

3.2 Sarcasm Detection

In this section, we first present some theories of sarcasm from a psychological perspective. Then, we survey the computational research methodologies employed for

detecting sarcasm from text documents. We categorize the existing techniques along the lines of non-affective (those not using any affective knowledge) and affective (exploiting affective information) models.

3.2.1 Theories of Sarcasm

Extensive research in psychology highlights a strong correlation between emotions and sarcasm. For example, recent findings indicate that performance on both emotion recognition and sarcasm detection decreases with increasing disease burden in Huntington disease (Larsen et al., 2016), the ability to understand sarcasm is dependent on emotion perception skills (Phillips et al., 2015), and that sarcasm intrinsically involves more emotion than literal language (Filik et al., 2015). It has also been suggested that sarcasm is one type of emotion (Riviello and Esposito, 2016) or more precisely, a language specific emotion (Chowdhuri and Bojewar, 2016), and that speakers may use it to show emotion and express surprise (Han, 2003). In addition, sarcasm has been shown to occur along different dimensions, emotion words being one of them (Campbell and Katz, 2012).

Furthermore, most theorists agree that sarcasm serves some communicative function that would not be achieved by speaking directly, such as eliciting a particular emotional response in the recipient (Filik et al., 2016). For example, one line of research suggests that the function of sarcasm is to mute the emotional impact of the message (Boylan and Katz, 2013; Jorgensen, 1996). In contrast, other researchers have found that the use of sarcasm actually enhances the emotional impact of the message (Colston, 1997; Kreuz et al., 1991). Although the existing evidence regard-

ing the perceived emotional impact of sarcasm compared to literal ones is mixed and conflicting, most researchers agree that sarcasm serves some kind of emotional function.

3.2.2 Non-affective Models

One of the earliest works in sarcasm detection from text mainly relied on lexical features such as n-grams (Kreuz and Caucci, 2007). A semi-supervised learning framework for recognition of sarcasm in Amazon product reviews (Tsur et al., 2010) and Twitter microblog posts (also called tweets) (Davidov et al., 2010) was proposed by exploiting syntactic and pattern-based features including high frequency words and content words, and punctuation-based features to build a weighted k-nearest neighbor classification model to perform sarcasm detection.

Previous studies have found that tweets with sarcasm hashtags are noisy and possibly biased towards the hardest form of sarcasm, where even humans have difficulty (Davidov et al., 2010), and predominantly tagged by certain types of Twitter users (Bamman and Smith, 2015), and therefore, are not reliable enough indicators to serve as ground truth data.

Filatova (2012) described the creation of a sarcasm corpus for Amazon product reviews, where the annotations captured sarcasm at document as well as text utterance level. Lukin and Walker (2013) explored the potential of a bootstrapping method for sarcasm classification in social dialogue to learn lexical n-gram cues associated with sarcasm (e.g., "oh really", "I get it", "no way", etc.) as well as lexico-syntactic patterns.

Liebrecht et al. (2013) also used n-gram features to detect sarcasm in Dutch tweets using a balanced winnow classifier and observed that people tend to be more sarcastic towards specific topics such as school, weather, returning from vacation, public transport, dentist, etc.

The role of additional context beyond the target text has also been explored in several studies (Bamman and Smith, 2015; Joshi et al., 2016a; Khattri et al., 2015; Rajadesingan et al., 2015; Wallace et al., 2014; Wallace, 2015; Wang et al., 2015). Wallace (2015) used meta-data about reddit comments; Rajadesingan et al. (2015) captured additional context related to the author, conversation, etc.

Interestingly, however, when it comes to humans, one study found that the annotators are *not* more likely to agree if given access to additional context beyond the target text and that the inter-annotator agreement was highest when only the target text was provided for annotation purposes (Abercrombie and Hovy, 2016).

A major limitation of most of these approaches is their use of extensive feature engineering which requires considerable time and effort. Thus, an alternative approach of inducing relevant features for sarcasm involves exploiting text representations learned automatically from large text corpora via neural network models, which help avoid the feature sparsity problem of discrete models (Ghosh and Veale, 2016; Ghosh et al., 2015; Joshi et al., 2017, 2016b; Zhang et al., 2016). Additionally, these neural features can capture long-range and subtle semantic patterns, which are difficult to express using discrete feature templates.

Most of the above-mentioned studies treat sarcasm as a linguistic and lexical phenomena, with limited emphasis on the affective aspects of sarcasm.

3.2.3 Affective Models

A recent thread of research in computational sarcasm detection explores the role of affective knowledge, in the form of binary sentiment categories such as positive and negative (Barbieri et al., 2014; Hernández-Farías et al., 2015; Joshi et al., 2015; Riloff et al., 2013), to more fine-grained categories of emotions such as *joy*, *disgust* and more (Farías et al., 2016; González-Ibáñez et al., 2011; Khokhlova et al., 2016; Poria et al., 2016; Reyes et al., 2012; Sulis et al., 2016).

The semi-supervised approach proposed by Riloff et al. (2013) used a sentiment lexicon to detect sarcasm based on the assumption that sarcastic tweets are a contrast between a positive sentiment and a negative situation. Barbieri et al. (2014) considered the amount of positive and negative words by using the sentiment lexicon SentiWordNet. Joshi et al. (2015) used features capturing explicit (overtly expressed through positive and negative words) and implicit (covertly expressed through phrases of implied sentiment) incongruity. Similarly, Hernández-Farías et al. (2015) exploited two sentiment lexicons as features in their model.

Some recent studies have also incorporated additional richer emotional information into models of sarcasm detection. For example, González-Ibáñez et al. (2011) introduced a sarcasm detection technique for tweets using numerous lexical features derived from LIWC (Pennebaker et al., 2001) and WordNet Affect (Valitutti, 2004), and pragmatic features (emoticons and replies) and discovered a strong correlation between positive and negative emotions and sarcasm. Reyes et al. (2012) developed classifiers to distinguish tweets containing the hashtag #irony from tweets containing the hashtags #education, #humour and #politics, based on ambiguity, polarity,

unexpectedness and emotional cues represented in terms of activation, imagery and pleasantness ratios (Whissell, 2009). More recently, Poria et al. (2016) investigated whether features from sentiment, emotion and personality models could improve sarcasm detection performance.

One study examining the distribution of emotions in sarcastic tweets found *trust* and *fear* to be the two most common emotions in sarcasm, while *surprise* and *joy* were the two least common emotions (Khokhlova et al., 2016). However, a different study examining the distribution of affective information found *joy* and *anticipation* to be the two most common emotions in sarcastic tweets, and *sadness* and *disgust* to be the least common (Sulis et al., 2016). Yet, another study highlighted the importance of the emotion *love* in discriminating sarcastic instances (Farías et al., 2016).

Most recently, word representations obtained from a corpus of tweets with noisy labels derived from emojis, have also been employed as features in computational sarcasm detection (Felbo et al., 2017).

In this thesis, we build upon these recent advancements highlighting the importance of affective features, by specifically studying the role of emotion information within sarcastic instances in two ways: incorporating affective knowledge into word representations in order to obtain richer features for detecting sarcasm (chapter 6); and, enhancing affective models for sarcasm detection by leveraging the transitions and relationships between different emotions within sarcastic utterances (chapter 7).

Chapter 4

Selective Co-occurrences for Word-emotion Association

Emotion classification from text typically requires some degree of word-emotion association, either gathered from pre-existing emotion lexicons or calculated automatically through unsupervised statistical corpus-based approaches of semantic relatedness. Most emotion lexicons contain a fixed number of emotion categories and provide a rather limited vocabulary coverage. Current measures of computing semantic relatedness, on the other hand, do not extend well to the specific task of word-emotion association and therefore, yield results that are far from satisfactory. In this chapter, we propose an unsupervised method of learning word-emotion association, called Selective Co-occurrences (SECO), from large unlabeled text corpora by leveraging certain properties generally exhibited by emotions. Extensive evaluation, using just one seed word per emotion category, indicates the effectiveness of the proposed approach over three manually created emotion lexicons and two state-of-the-art word

embedding models on three emotion classification datasets from diverse domains.

4.1 Introduction

Emotion detection at sentence or document level can be achieved in many ways. Underlying every approach, though, is the notion of word-level emotion association, which can also be obtained in a number of ways such as manual annotation or statistical corpus-based approaches.

Word-emotion association indicates the degree of association or relatedness between words and the different categories of emotions. For example, the word "*accident*" can be considered associated with the emotion *sadness*; the word "*birthday*" generally evokes the feeling of *joy* or *surprise*. Such association is typically obtained from a pre-compiled emotion lexicon or calculated automatically using statistics obtained from a large unlabeled corpora of text. The merits and demerits of each of these methods were discussed at length earlier in chapter 3.

While computing word-emotion association scores automatically through unsupervised statistical corpus-based approaches addresses the many limitations of utilizing manually annotated resources, the current models of semantic relatedness or word representations do not extend well to the specific task of emotion classification.

Unsupervised statistical approaches are primarily based on the concept of co-occurrences and context windows, i.e., words occurring together in similar contexts tend to be related. There are many ways of defining a context such as words only to the left of the target word, words only to the right of the target word, words on both sides of the target word, and so on. Moreover, the context window can extend over

multiple sentences within a document or be restricted to one sentence at most.

Consider the following piece of text:

The party was really fun, but we were sad that you couldn't join us.

In this example, most context-based models will consider the words 'party', 'fun' and 'sad' to be co-occurring together. However, ideally, only the words 'party' and 'fun' should be considered as co-occurring.

In this chapter, we seek to improve over the existing unsupervised statistical models of computing word-emotion association, especially the count-based models, by integrating the concept of *selective co-occurrences*.

Selective Co-occurrences (SECO) is motivated by inherent properties exhibited by emotions, namely mutual exclusivity (i.e., a word is largely associated with only one emotion category in any given context), and context weighting (i.e., words that appear nearer to each other are more strongly associated than words that appear farther away). By modifying the conventional co-occurrence-based methods, we compute a uni-directional asymmetric association between a given word and an emotion seed word. The proposed approach is found to be better at capturing the association between words and emotions than general purpose measures.

Extensive evaluation of word-emotion association scores derived from two large text corpora (Wikipedia articles and Amazon reviews), under the framework of unsupervised emotion classification on three emotion datasets from very diverse domains demonstrates the effectiveness of employing selective co-occurrences. The proposed approach is particularly interesting as it requires no labeled training data or manually created emotion lexicons, and can be extended to a flexible number of emotion

categories.

The remainder of this chapter is organized as follows. In section 4.2, we describe the learning of the word-emotion association scores and their use in the task of unsupervised emotion classification. Section 4.3 describes the evaluation setup, section 4.4 analyzes the experimental results, and section 4.5 presents model analyses. Lastly, section 4.6 concludes the chapter.

4.2 Word-emotion Association for Unsupervised Emotion Classification

Given a sentence s and a set $E = \{e_1, e_2, \ldots, e_g\}$ of g emotion categories, the objective is to label s with the best possible emotion $e_j \in E$. We first discuss our proposed method for deriving the word-emotion association scores between a word and an emotion category, and then use these scores to obtain an emotion label for each sentence in an unsupervised manner.

4.2.1 Learning Word-emotion Association

Let $W = \{w_1, w_2, \ldots, w_n\}$ be a set of n cue words in an input sentence s, where $W \subseteq s$. A cue word is defined as any word within a sentence that could have some emotion connotation. Usually, these are the nouns, adjectives, verbs and adverbs, excluding the stopwords. Assuming $E = \{e_1, e_2, \ldots, e_g\}$ to be the set of g emotion categories, an emotion category $e_j \in E$ is represented by one or more emotion seed words. Let $\mathcal{T}$ denote the set of all the seed words for all the emotion categories, and

$T_j = \{t_{j_1}, t_{j_2}, \ldots, t_{j_m}\}$, where $T_j \subset \mathcal{T}$ be the set of m emotion seed words for an emotion category $e_j \in E$.

As an illustration, consider the sentence "*Parties are fun*", where the set of cue words is $W = \{parties, fun\}$. If the classification scheme follows, for example, Ekman's (1992) model of emotions, then $E = \{anger, disgust, fear, happiness, sadness, surprise\}$, and the set of emotion seed words for the emotion *happiness*, for instance, could be $T_{happiness} = \{happy, joy, ...\}$ and *anger*'s seed words could include $T_{anger} = \{angry, mad, \ldots\}$.

We adopt the "$\rightarrow$" symbol to denote the association between a cue word w and an emotion seed word t. The first step involves deriving the association scores between a cue word and every seed word, e.g., $Assoc.(parties \rightarrow happy)$, $Assoc.(parties \rightarrow joy)$, $Assoc.(parties \rightarrow angry)$, and so on.

As our main goal is to acquire *association scores for emotion classification*, the design choices for our proposed measure of learning word-emotion association scores are largely motivated by the following observations:

- The intrinsic process of annotating an emotion dataset as well as that of classifying it is uni-directional, i.e., given a word or a sentence, the task is to label it with the emotion it evokes the most. Further, as noted by Tversky (1977), certain linguistic relationships are characteristically asymmetric. In one experiment to list the first meaningfully related word that comes to mind, for the cue word *fear*, 24% of the participants answered *scared*, while only 9% of them recalled *fear* when given the cue word *scared*, suggesting an inherent asymmetry in word associations (Altarriba et al., 1999).

- Although the expressions of emotions can sometimes be fuzzy, most words pri-

marily evoke only one emotion in a particular context, i.e., the emotion categories are, for the most part, mutually exclusive. As a matter of fact, in the emotion lexicon WordNet Affect (Strapparava and Valitutti, 2004), which contains words annotated with more than one emotion, about 98.7% of the terms are labeled with just one emotion.

- Most importantly, unlike other word relatedness tasks, the second half of the word-word association pair (i.e., emotion seed words) in this particular task are known in advance.

Traditional unsupervised statistical corpus-based methods (described earlier in section 3.1.3.2) indiscriminately consider all the words occurring within a context window as co-occurring. Furthermore, all the words are weighted equally, regardless of how far they appear from each other. In order to formulate a more suitable metric of word-emotion association, we integrate the following two properties into SECO.

1. **positional context and mutual exclusivity**: The concept of selective co-occurrences can be described as follows – a cue word is considered as co-occurring with only *one* emotion seed word within any particular window of text. Consider Figure 4.1 containing an example window of text, the cue word "*party*", and two emotion seed words "*angry*" and "*happy*", representing the two emotion categories, *anger* and *happiness*, respectively. When a context window contains multiple seed words from multiple emotion categories, three possible settings for selecting the most appropriate seed word as co-occurring with the cue word are explored:

> Theater critic Michael Riedel (playing himself) also shows up, uninvited. Ivy is put out by this and gets **angry** at Michael about it. We hear but don't see Ivy singing "Bittersweet Symphony" at her **party**. Derek then walks in and gives her a present and wishes her **happy** birthday.

Figure 4.1: Window of text containing cue (*party*) and seed (*angry, happy*) words

- *nearest* (`SECO-NEAR`): This is the most intuitive option, where the nearest seed word to the cue word is selected. For example, "*happy*" is counted as co-occurring with "*party*"; "*angry*" is ignored.
- *preceding* (`SECO-PREC`): To account for any positional predisposition, this setting considers only the closest preceding seed word to the cue word. For example, "*angry*" is considered as co-occurring with "*party*"; "*happy*" is ignored.
- *following* (`SECO-FOLL`): Similarly, this setting considers only the closest seed word that follows the cue word as co-occurring together. For example, "*happy*" is considered as co-occurring with "*party*"; "*angry*" is ignored.

2. **distance-based**: Words that appear nearer to each other have been found to exhibit stronger relationships (Beeferman et al., 1997). In the past, this property has been successfully exploited by incorporating a decaying factor which allows words that co-occur nearer to each other to be more related (Brosseau-Villeneuve et al., 2010; Gao et al., 2002; Mikolov et al., 2013a; Sahlgren, 2006). To this end, we also apply a context weighting scheme whereby a seed word is linearly weighted according to its distance from the cue word. In a window of

size k, the n^{th} word from the cue word is weighted by the following function:

$$\frac{k-n+1}{k} \tag{4.1}$$

For example, in a window of 5 words, the first word next to the cue word is weighted by $\frac{5}{5}$, while the fourth word away is of weight $\frac{2}{5}$. In other words, as the distance between two words increases, their weighted word-emotion association score decreases.

The selective counting of the seed word's co-occurrence frequency with the cue word essentially makes our association measure asymmetric and therefore, the order of the cue word and seed word in the association equation cannot be interchanged. That is, $Assoc.(w \rightarrow t)$ denotes the association between a cue word w and a seed word t, and $Assoc.(w \rightarrow t) \neq Assoc.(t \rightarrow w)$.

Selective counting can be explained as follows. Given a text corpus of size N, a cue word w and a seed word t, we would like to see how w is associated with t. Let $\#(w)$ and $\#(t)$ be the total number of occurrences of cue word w and the total number of occurrences of seed word t in the corpus, respectively. For each occurrence of w in the corpus, a window of size $2k$ centered at word w is considered. There are a total of $\#(w)$ such windows in the corpus. If, in such a window, t is the nearest, closest preceding or closest following seed word to cue word w, then t is considered to *selectively co-occur* with w. We use $\eta(w,t)$ to denote the total number of such selective co-occurences between w and t.

Technically, SECO is applicable to any traditional count-based co-occurrence word association measure that estimates the relatedness between two words by computing some function of the words' frequencies. In this chapter, we apply selective

co-occurrences to three popular co-occurrence association measures, namely NPMI (Bouma, 2009), Dice (1945) and Jaccard (1912).

- `SECO-NPMI`: The normalized `SECO-NPMI` between w and t, within the range of $[-1, 1]$, is:

$$SECO\text{-}NPMI\,(w \to t) = \frac{\log N \frac{\eta(w,t)}{\#(w)\#(t)}}{\log \frac{N}{\eta(w,t)}}. \tag{4.2}$$

- `SECO-Dice`: Similarly, `SECO-Dice` between w and t is computed as:

$$SECO\text{-}Dice\,(w \to t) = \frac{2\eta\,(w,t)}{\#\,(w) + \#\,(t)}. \tag{4.3}$$

- `SECO-Jaccard`: Lastly, one of the earliest co-occurrence associations measures, Jaccard, can be transformed as follows:

$$SECO\text{-}Jaccard\,(w \to t) = \frac{\eta\,(w,t)}{\#\,(w) + \#\,(t) - \eta\,(w,t)}. \tag{4.4}$$

Finally, the word-emotion association between w and an emotion category e_j is obtained by calculating the average mean of the association scores between w and all the seed words of e_j as follows:

$$Assoc.\,(w \to e_j) = \frac{1}{m}\sum_{i=1}^{m} Assoc.\,(w \to t_{j_i}) \tag{4.5}$$

where $Assoc.$ is any association measure such as `SECO-NPMI`.

4.2.2 Classifying Sentence Emotion

For each word w, its emotion vector ϕ_w is denoted as:

$$\phi_w = \langle Assoc.\,(w \to e_1)\,, Assoc.\,(w \to e_2)\,, \ldots, Assoc.\,(w \to e_g)\rangle \tag{4.6}$$

and the emotion vector ϕ_s of sentence s is obtained by averaging the emotion vectors of all its n cue words as follows:

$$\phi_s = \frac{1}{n} \sum_{i=1}^{n} \phi_{w_i}. \tag{4.7}$$

Finally, the sentence is labeled with the emotion category $e_j \in E$ with the maximum value in ϕ_s. Note that such an approach requires neither labeled set of training data, nor manually created emotion lexicons.

4.3 Evaluation Setup

In this section we describe the evaluation datasets, the text corpus used for learning the word-emotion association scores and the evaluation metric.

4.3.1 Evaluation Datasets

Below described are the three popular emotion datasets on which we evaluate the performance of our proposed approach. Some sample sentences from these datasets are presented in Table 4.1, while Table 4.2 summarizes their statistics.

Aman: Consisting of highly informal blog data, this dataset includes 1290 sentences annotated with one of six emotions: *anger, disgust, fear, joy, sadness* and *surprise* (Aman and Szpakowicz, 2007).

Alm: Emotions are particularly significant in the literary genre of fairy tales and this dataset contains 1207 high-agreement sentences (i.e., all four annotators agreed with the same emotion label) marked with one of the following five emotions: *angry-disgusted, fearful, happy, sad* and *surprised* (Alm, 2008).

Dataset	Sentence	Emotion Label
Aman	*Do you people not listen to the news or what?*	anger
	I had a blast in california hanging out with my family.	happiness
Alm	*Oh! cried the devil, "what are you doing?"*	surprised
	Ha! what are you doing? cried the devil angrily.	angry-disgusted
ISEAR	*When I saw a ghost.*	fear
	Slaughtering of animals.	disgust

Table 4.1: Sample sentences from evaluation datasets

ISEAR: Developed for studying the relationships among emotions and cultures, this corpus contains experiences evoking seven emotions: *anger, disgust, fear, joy, sadness, shame* and *guilt*, resulting in a total of 5412 sentences[1]. To the best of our knowledge, no existing lexicon contains *shame* or *guilt* emotion categories, and therefore, methods that exclusively depend on emotion lexicons for extracting word-emotion association cannot correctly classify sentences belonging to these emotions. However, unsupervised approaches such as ours, which can be initialized with as little as one seed word per emotion category, are easily applicable to such datasets.

4.3.2 Text Corpora

We derive the word-emotion association scores from the following two large unannotated text corpora originating from different domains:

[1]http://www.affective-sciences.org/system/files/webpage/ISEAR_0.zip.

	ag	*dg*	*fr*	*hp*	*sd*	*sp*	**Total**
Aman	179	172	115	536	173	115	1290
Alm	218		166	445	264	114	1207
ISEAR	1085	1072	1086	1089	1080	-	5412

Table 4.2: Statistics of evaluation datasets. *ag* denotes *anger*, *dg* is *disgust*, *fr* is *fear*, *hp* is *happiness*, *sd* is *sadness* and *sp* is *surprise*.

Wikipedia[2]: The large publicly available corpus of Wikipedia mainly consists of formal language structured text articles considered to be more "objective" in nature. Our clean corpus contains approximately 918.5 million tokens, with each article on one line.

Amazon: The text of all the product reviews, mostly consisting of informal language makes up our second corpus, considered to be of more "emotional" type. This data was extracted from the aggressively deduplicated dataset (McAuley et al., 2015), which contains 82.83 million product reviews from Amazon, spanning May 1996 - July 2014. Our clean corpus contains more than 3 billion tokens (three times the size of Wikipedia corpus), with one review per line.

All text is pre-processed by: (a) converting to lowercase; (b) stripping off all non-alphanumeric characters; (c) removing stopwords; (d) stemming[3], and (e) removing words that occur less than 5 times in the corpus.

[2]http://dumps.wikimedia.org/enwiki/20140811/enwiki-20140811-pages-articles.xml.bz2

[3]Stemming is the process of reducing inflected words to their word stem, base or root form. For example, the words 'walk', 'walks, 'walking', 'walked', all reduce to the stem 'walk'. We adopted the Porter stemmer for stemming: https://tartarus.org/martin/PorterStemmer/

4.3.3 Evaluation Metric

The results are evaluated in terms of F-score for each emotion class e, where F-score is the harmonic mean of *precision* and *recall*, defined as:

$$2\left(\frac{precision \times recall}{precision + recall}\right). \tag{4.8}$$

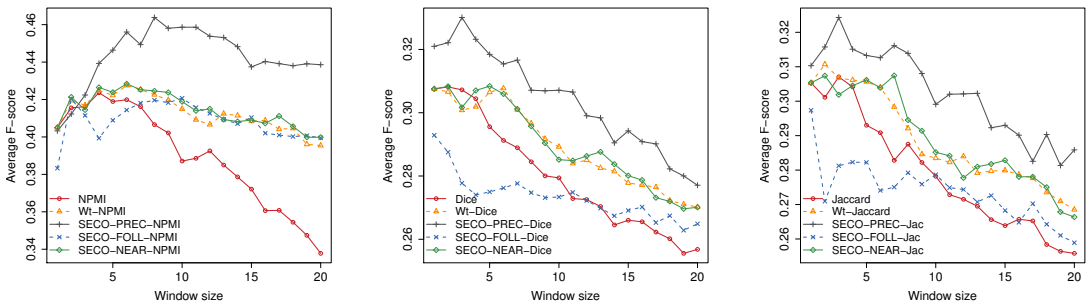
Precision is the number of sentences correctly labeled as belonging to the class e divided by the total number of sentences labeled as belonging to e, and *recall* is the number of sentences correctly labeled as belonging to e divided by the total number of sentences that actually belong to e. We report the results in terms of the average F-score over all the classes.

4.4 Experiments

In what follows, we evaluate the performance of the proposed approach, SECO, in several experiments and discuss their results.

4.4.1 How Effective is Selective Co-occurrence?

We test the performance of the three flavors of selective co-occurrences (`SECO-NEAR`, `SECO-PREC` and `SECO-FOLL`) to three traditional measures of semantic relatedness (NPMI, Dice, Jaccard), against regular (i.e., without selective co-occurrence, e.g., NPMI) as well as weighted regular versions (i.e., where the same context weighting scheme as described in Section 4.2.1 is applied to the regular association measures, e.g., Wt-NPMI) in order to analyze the effect of selective co-occurrence in particular, and not just the advantage obtained using weighted contexts.



(a) NPMI (b) Dice (c) Jaccard

Figure 4.2: Results of regular, weighted regular and selective co-occurrences on Alm dataset

As observed from Figure 4.2, tested for context window sizes 1 to 20 and trained on the Wikipedia corpus, our proposed approach, *preceding* selective co-occurrences (`SECO-PREC`), where only the nearest seed word that precedes a given cue word within a context window is considered as co-occurring, exhibits the best performance when applied to all the three association measures (NPMI, Dice and Jaccard), on the Alm dataset[4]. In fact, the best average F-score from `SECO-PREC-NPMI` is almost 10% better than that of Wt-NPMI, leading us to conclude that the gain in performance is due to selective co-occurrences and not just weighted contexts. As expected, the weighted versions, Wt-NPMI, Wt-Dice and Wt-Jaccard perform much better than their regular unweighted counterparts, NPMI, Dice and Jaccard, respectively.

Furthermore, `SECO-NEAR` shows a slight advantage over weighted association measures, while `SECO-FOLL` and regular association measures (i.e., NPMI, Dice, Jaccard) are nearly always the poorest performing. We suspect that `SECO-PREC`'s superior

[4]Consistent results were obtained on the Aman and ISEAR datasets.

performance over SECO-NEAR and SECO-FOLL could be due to the fact that most emotion seed words belong to the adjective class which, in English language, are found preceding other types of words. Since SECO-PREC-NPMI yielded the best overall results, we further evaluate its performance against state-of-the-art baselines in the next experiment.

4.4.2 How Effective Is SECO-PREC-NPMI?

In this experiment, we evaluate the performance of unsupervised SECO-PREC-NPMI against five baselines (indicated in **bold**) described next.

4.4.2.1 Baselines

Emotion lexicons including WordNet Affect (**WNA**), NRC Emotion Lexicon (**EmoLex**) and DepecheMood (**DM**) (described earlier in section 3.1.3.1) contain words and their association with various emotions.

For each emotion category, WNA contains a simple list of words, which we interpret as a binary association; if a word exists in an emotion category, we assign +1 for that emotion.

EmoLex and DM, on the other hand, contain association scores between a word and all the emotion categories. For instance, EmoLex lists the association between the word "*awful*" and 8 emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust) as: $1, 0, 1, 1, 0, 1, 0, 0$. In DM, each word is associated with a different set of emotions by a real valued score, summing upto 1. For instance, the association between the word "*awe*" and 8 emotions (afraid, amused, angry, annoyed, dont_care, happy, inspired, sad) is listed as: $0.08, 0.12, 0.04, 0.11, 0.07, 0.15, 0.38, 0.05$. We use

these emotion lexicons as a baseline by applying a keyword matching algorithm, to obtain $Assoc.(w \rightarrow e)$. For example, $Assoc.(awe \rightarrow sad) = 0.05$ from DM. Note that, since DM does not contain two of the emotion categories found in our evaluation datasets, i.e., *disgust* and *surprise*, we report its results on a subset of the datasets. Instead of directly comparing our word-emotion association scores with those of emotion lexicons, we evaluate them under the framework of emotion classification as there are significant differences between the various emotion lexicons and none can be considered to be a perfect benchmark.

Semantic similarity scores computed from two state-of-the-art word embedding algorithms, continuous bag-of-words (**CBOW**) and skip-gram (**SG**) (Mikolov et al., 2013a,b) (described earlier in section 3.1.3.2), offer another relevant baseline as they can be used to obtain unsupervised word-emotion association scores, which is closer in spirit to our goal. We used the algorithms' recommended default parameter settings: *dimension size of feature vectors* = 300; *negative sampling* = 5, with the rest of their setup for learning word-emotion association (e.g., training corpus, window size, etc.) similar to that of SECO.

Unlike SECO, which directly outputs association or similarity scores between two words, word embedding models such as CBOW and SG produce word vectors, which are word representations in some multidimensional space. Assume a and b to be the two words for which we wish to compute the association score. For example, a could be a cue word and b could be a seed word. Let $\mathbf{a}$ and $\mathbf{b} \in \mathbb{R}^d$ denote the corresponding word vectors of a and b, respectively, in a d-dimensional vector space. Then, the

association between a and b is computed through cosine similarity as follows:

$$\text{cosine}\,(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|\,\|\mathbf{b}\|} \tag{4.9}$$

where $\|\mathbf{a}\|$ is the l_2-norm of the vector, and $\mathbf{a} \cdot \mathbf{b}$ is the dot product of the two word vectors.

Since the context window size can have a significant impact on the performance of an algorithm, we run each method of semantic relatedness (SECO, CBOW, SG) on 20 different window sizes (1 to 20) on both the text corpora (Wikipedia and Amazon) and report the average result with standard deviation for each setting in Table 4.3. To keep the process as unsupervised as possible, in this study only one seed word per emotion category is used to guide the learning of the association scores. The seed words "*angry*", "*disgust*", "*happy*", "*scared*", "*sad*" and "*surprise*" represent the six emotion categories, *anger, disgust, happiness, fear, sadness* and *surprise*, respectively.

4.4.2.2 Results

In particular, the following observations are made:

(i) Usually it is difficult to determine the best window size in advance, therefore, for window sizes 1 to 20, we summarize the average F-scores over all the window sizes in Table 4.3. The results indicate that in general, `SECO-PREC-NPMI` yields better overall results, suggesting the effectiveness of selective co-occurrences in this task. with $\text{SG}_{\text{amazon}}$ obtaining competitive results on one dataset (Aman). Interestingly, contrary to popular intuition, the "objective" text from the Wikipedia training corpus yields better F-scores on average than the "subjective" Amazon reviews corpus for two out of the three evaluation datasets.

	Aman	**Alm**	**ISEAR**
SG_{wiki}	0.242 ± 0.04	0.209 ± 0.05	0.259 ± 0.08
$CBOW_{wiki}$	0.382 ± 0.02	0.426 ± 0.03	0.446 ± 0.03
SECO-PREC-$NPMI_{wiki}$	**0.410** ± 0.01**	**0.443** ± 0.01**	0.488 ± 0.02**
SG_{amazon}	**0.410** ± 0.02*	0.406 ± 0.02	0.438 ± 0.04
$CBOW_{amazon}$	0.393 ± 0.02	0.373 ± 0.02	0.484 ± 0.03
SECO-PREC-$NPMI_{amazon}$	0.403 ± 0.01	0.409 ± 0.01**	**0.498** ± 0.02**

Table 4.3: Average F-scores (of windows 1 to 20) for three evaluation datasets. SG, CBOW and `SECO-PREC-NPMI` were run on Wikipedia and Amazon corpora for windows 1 to 20. The best average result for each dataset is in **bold**. $^{**}p < .00001$, $^{*}p < .01$ (one-way ANOVA test for each dataset results using the same training corpus, i.e., wiki or amazon)

(ii) The best results for each method as compared with the emotion lexicons are further detailed in Table 4.4. The results indicate that, on all the three evaluation datasets, with just one seed word per emotion category used to derive the word-emotion association scores, all the unsupervised measures of semantic relatedness (`SECO-PREC-NPMI`, SG and CBOW) outperform all the emotion lexicons (WNA, EmoLex and DM) that were created using considerable human input and training data, indicating that semantic similarity approaches provide an effective unsupervised way of extracting meaningful word-emotion association scores. Within the emotion lexicons, WNA provides the best performance on two out of the three datasets despite being the smallest in size. Unsupervised association measures demonstrate

	Aman	**Alm**	**ISEAR**
WNA	0.286	0.362	0.343
EmoLex	0.316	0.341	0.318
DM	0.324	0.340	0.290
SG_{wiki}	0.338 (2)	0.345 (1)	0.433 (1)
$CBOW_{wiki}$	0.410 (15)	0.456 (11)	0.481 (19)
SECO-PREC-$NPMI_{wiki}$	0.422 (10)	**0.464** (8)	0.497 (10)
SG_{amazon}	**0.435** (6)	0.440 (1)	0.490 (5)
$CBOW_{amazon}$	0.411 (6)	0.399 (18)	0.510 (19)
SECO-PREC-$NPMI_{amazon}$	0.412 (11)	0.422 (15)	**0.512** (20)

Table 4.4: Details of best results for three evaluation datasets. The best result for each dataset is in **bold**. The window size is shown in parentheses.

two significant advantages over emotion lexicons: first, association measures are able to provide a wider coverage by exploiting the inherent associations between words that are present in text corpora; and second, while the lexicons contain fixed predetermined categories of emotion, association measures can be flexibly extended to any number and types of emotions. As for the recommended window settings for each approach, it seems that SG works well with window size = 1 on Wikipedia corpus and around 5 on Amazon; CBOW usually does well on windows larger than 15 words and `SECO-PREC-NPMI` is recommended to be used with window size of 10 words on Wikipedia and larger than 15 on Amazon.

(iii) In order to further analyze the results of each individual emotion category,

AMAN							
	ag	*dg*	*fr*	*hp*	*sd*	*sp*	Avg
SG_{az}	0.36	**0.50**	**0.45**	0.47	**0.44**	**0.34**	0.435
$CBOW_{az}$	0.38	0.49	0.33	0.53	0.38	0.32	0.411
$SECO_{wk}$	**0.43**	0.30	0.38	**0.65**	0.43	**0.34**	0.422

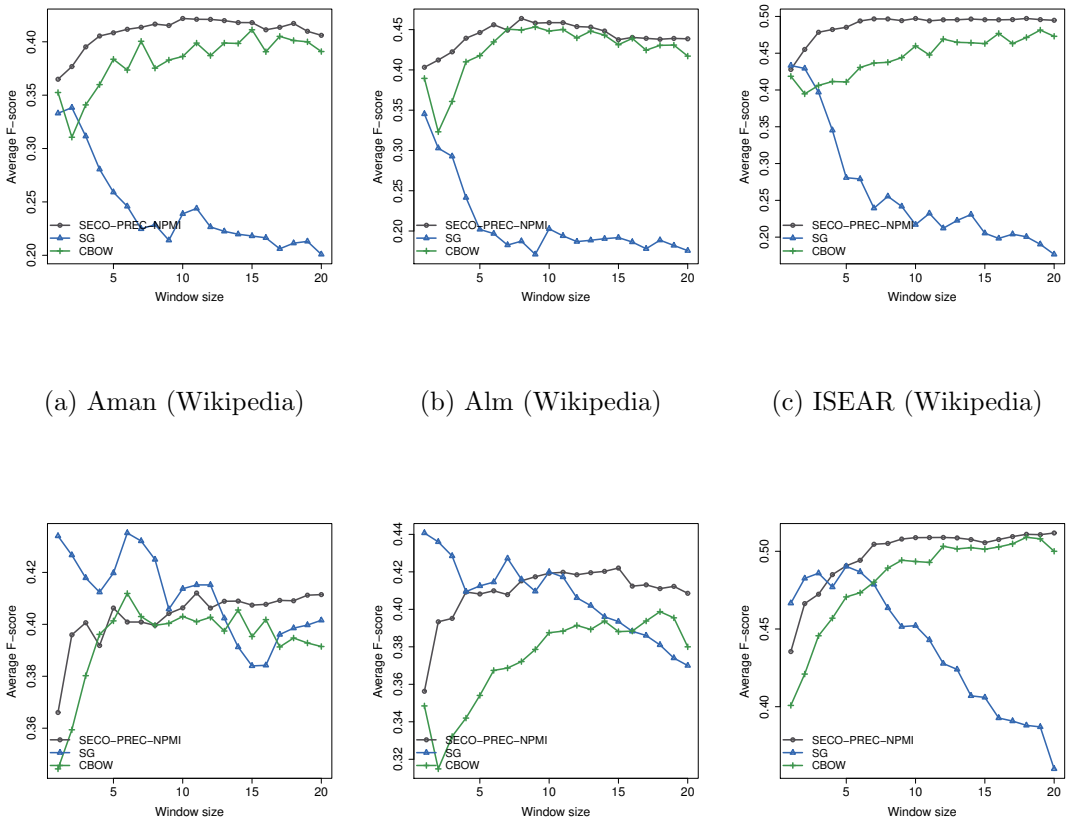
ALM						
	ag-dg	*fr*	*hp*	*sd*	*sp*	Avg
SG_{az}	**0.50**	0.46	0.44	0.51	0.27	0.440
$CBOW_{az}$	0.47	0.43	0.44	0.47	0.14	0.399
$SECO_{wk}$	0.36	**0.47**	**0.67**	**0.55**	**0.28**	0.464

ISEAR						
	ag	*dg*	*fr*	*hp*	*sd*	Avg
SG_{az}	0.38	**0.56**	0.55	**0.47**	0.46	0.490
$CBOW_{az}$	0.50	0.53	**0.59**	0.42	**0.49**	0.510
$SECO_{az}$	**0.51**	**0.56**	**0.59**	0.42	0.48	0.512

Table 4.5: Details of emotion category results for best window size/training corpus combination for three evaluation datasets. *ag* = anger, *dg* = disgust, *fr* = fear, *hp* = happy, *sd* = sad, *sp* = surprise.

we present the F-scores of the best approach/training corpus combination in Table 4.5. In general, the *happiness* category obtains the highest results in two datasets while *fear* does best on the third. On the other hand, the most difficult category to be classified correctly seems to be *surprise*. One avenue of future work could include experimenting with various seed words as a means of increasing the accuracy of such emotions.

(iv) A deeper analysis of the results revealed that in almost all the datasets, the most misclassified instances belonged to the *happiness* class. This could be because



(a) Aman (Wikipedia) (b) Alm (Wikipedia) (c) ISEAR (Wikipedia)

(d) Aman (Amazon) (e) Alm (Amazon) (f) ISEAR (Amazon)

Figure 4.3: Parameter sensitivity results for 20 window sizes.

happiness seems to be expressed by a larger vocabulary (as indicated by the number of words under the *happiness* category in most lexicons, as well as the number of *happy* instances in all the evaluation datasets). Similarly, the second most misclassified instances belonged to the *sadness* class, with most of them incorrectly labeled as *happiness*. This issue is especially challenging as illustrated by the following mislabeled sentence: "*I'm losing enjoyment out of things I love to do and that's never a good sign*", which calls for a more comprehensive emotion classification algorithm.

In conclusion, the results can be summarized as follows:

- Initialized using as little as one seed word per emotion category, the measures of semantic relatedness outperform the emotion lexicons which require considerable time and effort.

- Amongst the measures of semantic relatedness, the proposed approach of applying selective co-occurrences, SECO, considerably outperforms regular measures of relatedness, suggesting the importance of positional context, mutual exclusivity and context weighting for obtaining word-emotion association scores.

- When the window size is not known, in general, `SECO-PREC-NPMI` yields consistent promising results on all the evaluation datasets, while SG provides competitive results on one dataset.

- `SECO-PREC-NPMI` and CBOW yield better results with larger window sizes, whereas SG is best on windows less than 5 words.

4.5 Model Analysis

4.5.1 Effect of Context Window Size

As noted earlier, the context window size can have a significant impact on the performance of an algorithm. To analyze the effect of different window sizes, Figure 4.3 summarizes the results with respect to context window size sensitivity of the three algorithms, SG, CBOW and `SECO-PREC-NPMI`. On Wikipedia corpus, `SECO-PREC-NPMI` is consistently better than the others, whereas SG takes better advantage of the

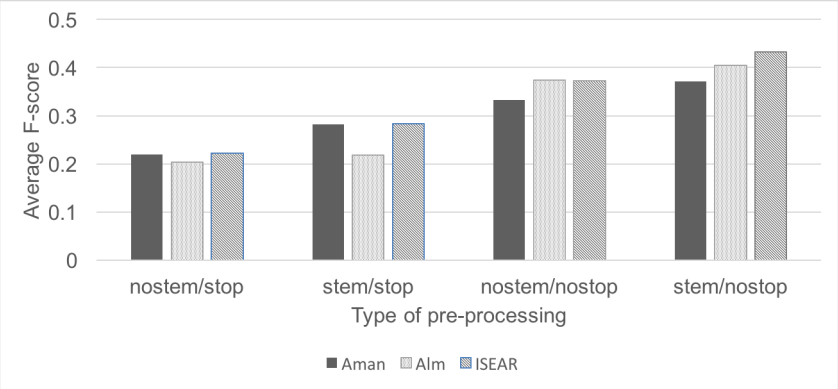


Figure 4.4: Effect of type of pre-processing. *stem* denotes stemming; *nostem* denotes no stemming; *stop* indicates stopwords not removed; *nostop* indicates stopwords removed.

Amazon corpus. While `SECO-PREC-NPMI` and CBOW get better with bigger context windows, SG depicts the opposite trend, best on windows less than 5 words.

4.5.2 Effect of Type of Pre-processing

The text data of the underlying corpus used for learning the word-emotion association scores was pre-processed by applying stemming and removing stopwords (described earlier in section 4.3.2). In Figure 4.4, we plot the results for three datasets obtained with the different types of pre-processing. For all the three datasets, significant improvement is observed under the *stem/nostop* option, i.e., stemming is applied and stopwords have been removed.

4.6 Conclusions

Emotion detection from text requires the degree of word-emotion association which is generally obtained from emotion lexicons or statistical corpus-based measures of semantic relatedness. While manual lexicons require considerable human time and effort, the current automatic techniques yield poor performance. In this chapter, we described a novel approach to automatically learning word-emotion association scores from unlabeled large text corpora. Evaluated under the framework of unsupervised emotion classification, initialized with just one seed word per emotion category, our proposed approach `SECO-PREC-NPMI` significantly outperformed three emotion lexicons and two state-of-the-art word embeddings models when trained using the Wikipedia text corpus.

Chapter 5

Learning Emotion-enriched Word Representations

Most word representation models consider the lexical semantics and syntax based on the co-occurrences of words. As a consequence, emotionally dissimilar words such as "*happy*" and "*sad*" occurring in similar contexts are estimated to be more similar than the emotionally similar word pair "*happy*" and "*joy*", which leads to rather undesirable consequences in affective tasks, such as emotion classification. In order to address this limitation, we propose a novel method of obtaining emotion-enriched word representations, which projects emotionally similar words into neighboring spaces. The proposed approach leverages distant supervision to automatically obtain a large training dataset of text documents and two recurrent neural network architectures for learning the emotion-enriched representations. In extensive evaluation on two tasks including emotion classification and emotion similarity, the proposed representations outperform several competitive generic as well as affective word embeddings.

5.1 Introduction

Despite its potentially wide-spread use, the automatic detection of emotions remains a challenging multi-class, sometimes multi-label, classification problem due to a number of reasons, including: (i) different emotion models consist of different number and types of emotion categories; (ii) emotions are not only subjective but also fuzzy, with more than one emotion possible at the same time. As a result, development of emotion related resources, such as training data, has been limited to a few manually annotated datasets or lexicons, a process that requires much time and effort, and is expensive.

Automatically inferring word-emotion association from large unlabeled bodies of text through statistical corpus-based approaches provides many benefits such as wider vocabulary coverage, requiring little manual effort, and so on. Essentially, such approaches are based on the distributional hypothesis which states that words occurring in similar contexts are related.

Statistical approaches can be categorized along the lines of count-based (e.g., LSA) or predictive models (e.g., neural probabilistic language models) (Marco Baroni, 2014). Count-based methods compute the statistics of how often some word co-occurs with its neighbor words in a large text corpus, whereas, predictive models directly try to predict a word from its neighbors.

One such count-based method for learning word-emotion association in an unsupervised manner, SECO, was introduced in the previous chapter. Regardless of the numerous benefits of unsupervised learning (e.g., not requiring any labeled data, applicable to flexible taxonomies of emotions), one limitation of such approaches is

their average performance. In this chapter, we present a novel predictive model for learning emotion-enriched word representations, with the aim of further improving the results of emotion classification.

Word embeddings can be generated in one of two ways: unsupervised learning or semi-supervised learning. Most word embeddings assuming the unsupervised learning paradigm (e.g., skip-gram, CBOW, GloVe, etc.) are typically modeled using the context of words following the distributional hypothesis, i.e., words which occur in similar contexts tend to be semantically similar. As an illustration, consider the following sentence:

the quick brown fox jumped over the lazy dog

Assuming the skip-gram model, which predicts the context words given a target word in a window of size $2k + 1$ (i.e., k words to the left as well as k words to the right), and assuming $k = 1$, the following context windows containing the target word in the center will be generated:

the quick brown

quick brown fox

brown fox jumped

...

and the training data comprising of input (i.e., target word) and output (i.e., context word) variables for the predictive model will look like:

(quick, the), (quick, brown), (brown, quick), (brown, fox), ...

Word Pairs	GloVe	CBOW
(happy, joy) ⇑	0.601	0.355
(happy, sad) ⇓	0.643	0.535
(cry, weep) ⇑	0.605	0.574
(cry, laugh) ⇓	0.657	0.403

Table 5.1: Cosine similarity between emotionally similar (⇑) and emotionally dissimilar (⇓) word pairs

While such unsupervised learning has several benefits such as not requiring any labeled data as it relies only on the neighboring context words, it also poses a severe disadvantage in an affective task such as emotion classification where emotionally *dissimilar* words with similar contexts get mapped into neighboring spaces. To further motivate this limitation, Table 5.1 presents the cosine similarity between the word vectors of a few word pairs obtained from popular pre-trained unsupervised word embeddings such as GloVe (Pennington et al., 2014) and CBOW (Mikolov et al., 2013b). According to the similarity scores, both GloVe and CBOW rate the word pair (*happy*, *sad*) as more similar than (*happy*, *joy*).

The effectiveness of unsupervised word embeddings can be enhanced by infusing a small amount of human expertise under the paradigm of semi-supervised learning. In this chapter, we propose learning emotion-enriched word representations[1], which we call Emotion Word Embeddings (EWE), through a semi-supervised setup in order to project emotionally similar words into neighboring spaces. Towards that end, first,

[1]Available upon request.

a method of distant supervision is employed to automatically create a large training dataset with a rich spectrum of emotions. Then, two recurrent neural network architectures are developed to learn emotion-aware word representations by leveraging the noisy, but large training data. Specifically, we use Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) to capture the *contextual* information between the words of the text document as well as the *emotion* information provided in the form of the target label obtained through distant supervision. Experimental evaluation demonstrates the effectiveness of learned emotion embeddings in the two tasks of emotion classification and emotion similarity.

The major contributions of this chapter include:

- a novel distant supervision method for automatically labeling a large corpus of training data with fine-grained emotions;
- two LSTM model architectures for learning emotion-enriched word embeddings from text documents (a single-label model and a multi-label model);
- a detailed evaluation of the learned word vectors on two tasks: emotion classification over four domains of text (blogs, fairy tales, personal experiences, and tweets) and emotion similarity;
- a qualitative analysis of the learned emotion embedding space;
- and, a comparative evaluation of the model proposed in this chapter (EWE) with the one proposed in the previous chapter (SECO).

The rest of this chapter is organized as follows. The details of our proposed model, EWE, for learning emotion-aware word representations are presented in section 5.2.

Next, the experimental setup and the results are discussed in section 5.3, and section 5.4 presents the qualitative analysis. Then, section 5.5 compares the enhanced models of emotion connotation presented in this chapter against those discussed previously in chapter 4. Finally, section 5.6 concludes the chapter.

5.2 Emotion-enriched Word Representations

First, we explain the two neural network models and their components for learning Emotion Word Embeddings (EWE) in section 5.2.1. Essentially, the models are used to learn word embeddings (which are the weights of the connections between the input and the hidden layers) through supervised learning with labeled data. Then, we describe the process of automatically obtaining a large training dataset of text documents labeled with emotion through distant supervision in section 5.2.2.

5.2.1 Training Word Embeddings using LSTM

Let $\mathcal{V} = \{w_1, ..., w_{|\mathcal{V}|}\}$ denote the vocabulary of word tokens in the annotated dataset. Each word $w_i \in \mathcal{V}$ is represented as an n-dimensional continuous vector $\mathbf{x}_i \in \mathbb{R}^n$ and the full embedding matrix is $E \in \mathbb{R}^{n \times |\mathcal{V}|}$. Starting from original embeddings $\mathbf{x}_i^o$ of word w_i (initialized either randomly or through some pre-trained word embeddings), the goal is to learn emotion-enriched embeddings $\mathbf{x}_i^e$ for w_i.

The LSTM (Long Short-Term Memory) model finds a dense low dimensional representation of words by sequentially and recurrently processing each word in a document. Specifically, the inputs of the LSTM are preprocessed text documents that consist of a sequence of words and their corresponding target variable. Let

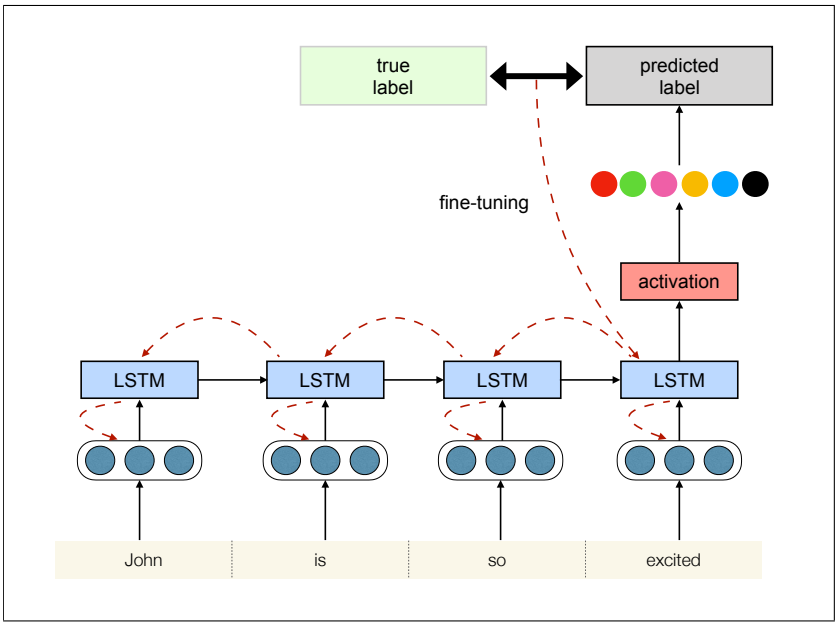


Figure 5.1: Learning word representations through LSTM model

$\mathcal{D} = \{(d_1, y_1), ..., (d_D, y_D)\}$ denote an annotated dataset of documents, where $d_i = \{w_1, w_2, ..., w_N\}$ denotes a text document consisting of a sequence of N words and y_i is the corresponding emotion label distribution for document d_i. The words of the text document are, first, converted into vector representations, which are then sequentially fed into the LSTM model left-to-right. Then, through back-propagation, the original word vectors get updated during training, producing emotion-enriched embeddings $\mathbf{x}_i^e$ for all $w_i \in \mathcal{V}$. An overview of the word representation learning through the LSTM model is presented in Figure 5.1.

In this chapter, we consider two model architectures to capture the *context* information by modeling the long-range dependencies between the words of a text document and *emotion* information provided through the target label to map each word into an affective embedding space. Model 1 (EWE_{UNI}) considers a single emo-

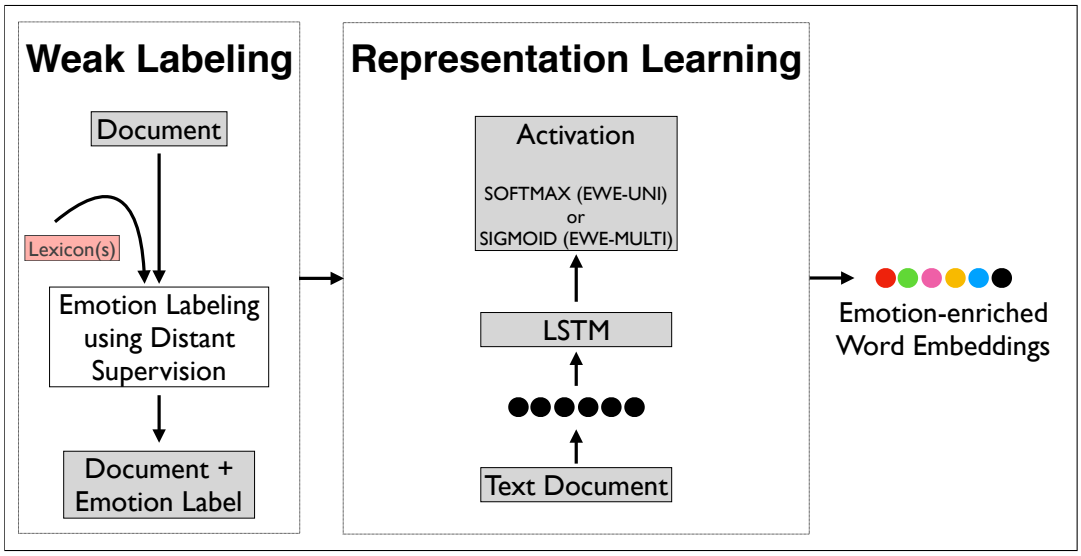


Figure 5.2: Overview of the framework for obtaining emotion-aware word representations

tion label for each document, whereas Model 2 ($\text{EWE}_{\text{MULTI}}$) allows multiple labels for a document. Figure 5.2 presents an overview of the proposed framework. First, we create a corpus of emotion-labeled documents using emotion lexicons through a distant supervision process (to be described in Section 5.2.2). Then this corpus is used as training data to learn emotion-enriched word representations using LSTM.

5.2.1.1 Model 1: EWE_{UNI}

Most words evoke only one emotion depending on the context. As an example, consider two benchmark emotion datasets (Alm et al., 2005; Aman and Szpakowicz, 2007) where each sentence is annotated with a single emotion label. Guided by this intuition, we propose EWE_{UNI} which follows a multi-class setting, where there exists only one valid mutually exclusive emotion label l_i for a text document d_i, and $l_i \in \mathcal{L}$, where $\mathcal{L} = \{l_1, ..., l_k\}$ denotes a discrete, finite set of k emotions.

Given an annotated document with its associated emotion label, the target value y is a one-hot vector, where the values of all the indices but one are 0. For example, if d is labeled with emotion l_i, then it holds that:

$$y_j = \begin{cases} 1, & \text{if } y_j = l_i \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

The neural network consists of one hidden layer, with the embedding matrix E added to the input layer. To predict the emotion label of the input text, an output layer with a softmax activation function which gives a probability distribution over the k classes is added on top of the hidden layer for modeling multi-class probabilities. The softmax function converts the classification result into label probabilities, i.e. $y' \in [0,1]^k$.

The final training objective is to minimize the multinomial cross-entropy loss of the predicted and true distributions, where the error over a batch of n documents is calculated as:

$$\mathcal{L} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} y_{ij} \log \left(y_{ij}^{'} \right) \tag{5.2}$$

where i denotes the ith training sample, j denotes the jth class, y is the true distribution (one-hot representation of the emotion label), and y' is the predicted probability distribution, $y'_{ij} \in [0,1]$ and $\sum_j y'_{ij} = 1$.

5.2.1.2 Model 2: $\text{EWE}_{\text{MULTI}}$

Although modeling emotion classification as a multi-class problem captures the basic emotion connotation of many words, in reality, several words can be associated with more than one emotion. For instance, during the process of creating the NRC EmoLex

emotion lexicon (Mohammad and Turney, 2013), it was found that *anger* words tend to be associated with *disgust* words, *joy* terms tend to be related with *trust*, and *surprise* terms are largely also associated with *joy*.

In order to capture a word's association with more than one emotion, the $\text{EWE}_{\text{MULTI}}$ models multi-class multi-label classification setup where each document can belong to multiple emotion classes at the same time. Assuming k emotion classes, and more than one valid emotion label for each document, the target variable y is binary represented. In other words, $y_j = 1$ indicates presence of an emotion class, and $y_j = 0$ otherwise. For example, if document d_i is labeled with a subset of emotion classes $s_i \subseteq \mathcal{L}$, then:

$$y_j = \begin{cases} 1, & \text{if } y_j \in s_i \\ 0, & \text{otherwise} \end{cases} \tag{5.3}$$

To predict the emotion label of the input text, an output layer with a sigmoid activation function, which squashes the inputs into a probability range of $[0, 1]$ for every class, is added to the last layer for modeling the probability of each class independently from the other classes.

The loss objective in this case is binomial cross-entropy, computed as follows:

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log\left(y'_i\right) + \left(1 - y_i\right)\log\left(1 - y'_i\right)\right] \tag{5.4}$$

where i denotes the ith training sample, y is the binary representation of true emotion label, and y' is the predicted probability.

5.2.1.3 Implementation

We use pre-trained word embeddings from GloVe, $|\mathcal{V}| = 1.9$ million, $n = 300$ (Pennington et al., 2014), whose effectiveness over other published embeddings has been previously explored (Ma and Hovy, 2016) to initialize E and use random initialization sampled from a zero mean Gaussian distribution: $x \sim \mathcal{N}(0, \sigma^2)$ for words not found in the pre-trained embeddings. Optimization of the loss function is carried out with the Adam optimizer (Kingma and Ba, 2014), which is known for yielding quicker convergence, with learning rate of 0.001, and mini-batch size set to 1024. The program is implemented in Keras[2] (Chollet et al., 2015) with Theano backend[3] (Theano Development Team, 2016) and trained on Nvidia Tesla K40c GPU.

5.2.2 Labeling Training Data using Distant Supervision

To learn the emotion embeddings, we require a large dataset of text with corresponding emotion labels. As it is quite challenging to create a sizable manually labeled training dataset due to human time and effort required, we leverage distant supervision (Go et al., 2009) to create a weakly labeled training dataset automatically in order to learn emotion-enriched word representations for a much larger vocabulary. Distant supervision is the process of labeling instances based on some heuristics or rules, with some of the instances being assigned noisy or imprecise labels.

[2]https:// github.com/ fchollet/keras

[3]http://www.deeplearning.net/software/theano/

5.2.2.1 Distant Supervision for $\mathbf{EWE_{UNI}}$

Let $\mathcal{D} = \{d_1, d_2, ..., d_D\}$ be the set of unlabeled documents. The goal is to generate an annotated dataset $\mathcal{D} = \{(d_1, l_1), ..., (d_D, l_D)\}$, where $l_i \in \mathcal{L}$ is the corresponding emotion label for document d_i and $\mathcal{L} = \{l_1, ..., l_k\}$ is a known finite set of emotion labels.

Let $d = \{w_1, w_2, ..., w_{|d|}\}$ denote the sequence of words in a document, $w_i \in d$. For each word w_i, we compute an emotion vector $\mathbf{a}(w) = \langle a_1, a_2, ..., a_k \rangle$, where a_j indicates the word-emotion association as derived from a lexicon. Although technically any emotion taxonomy can be followed for deriving the word-emotion vector $\mathbf{a}(w)$, in this chapter, we adopt Ekman (1992) model of six emotions (*anger*, *disgust*, *fear*, *happiness*, *sadness* and *surprise*), whose origins are firmly grounded and extensively verified in psychology. To this end, we select **WordNetAffect (WNA)** (Strapparava and Valitutti, 2004), which was developed by manually labeling the emotions of a few seed words and extending it to all their WordNet synonyms, and **NRC EmoLex (NRC)** (Mohammad and Turney, 2010, 2013), which was created through crowdsourcing by annotating unigrams with one or more of Plutchik (1980) eight emotions, which in turn is a superset of Ekman's model. In WNA, each word is associated with only one emotion, therefore $a_j = 1$ if w is associated with that emotion, and $a_j = 0$ otherwise. On the other hand, in NRC, a word can be binary associated with more than one emotion, with 1 indicating an association and 0 denoting no association. For a given word w, we extract its binary association scores corresponding to the six categories of Ekman's model.

The emotion vector $\mathbf{a}(d)$ for document d is then, the sum of the emotion vectors

of all its words, $\mathbf{a}(d) = \sum_{i \in d} \mathbf{a}(w_i)$. If the document has an association with at least one emotion, i.e., $\exists j, a_j(d) > 0$, then, $S = \text{argmax}_i \ \mathbf{a}(d)$, where $S \subseteq \mathcal{L}$. In other words, documents assigned zero emotion score are not considered. Finally, in case multiple emotion labels have the maximum value, i.e., $|S| > 1$, we sample uniformly at random one emotion label $l \in S$.

We investigate two strategies of computing the affective knowledge:

(i) **one lexicon** - where only one lexicon is used to guide the labeling process;

(ii) **two or more lexicons** - whereby two or more lexicons are leveraged in order to mitigate some effects of noisy labeling. This variant assigns an emotion label to a document only if the labels output by both the lexicons match.

5.2.2.2 Distant Supervision for $\text{EWE}_{\text{MULTI}}$

Some words evoke more than one emotion at the same time. For example, out of the 14,000 words annotated with emotions in the NRC lexicon, almost 8,000 words (57%) are associated with more than one emotion. Therefore, we relax the labeling scheme followed in EWE_{UNI} and design $\text{EWE}_{\text{MULTI}}$ to take into consideration a multi-class, multi-label setting, where a document can have more than one emotion label.

Unlike EWE_{UNI}, in $\text{EWE}_{\text{MULTI}}$ the set of all emotions with $a_j(d) > 0$ for document d is used as final emotion labels for d. Thus, the multi-label annotated dataset $\mathcal{D}$ is $\{(d_1, S_1), ..., (d_n, S_n)\}$, where document d_i is assigned a set of emotion labels $S_i \subseteq \mathcal{L}$

5.2.2.3 Training Data

Our large corpus of unlabeled documents is extracted from the Amazon reviews dataset (McAuley et al., 2015) consisting of product reviews, spanning May 1996

Review	Emotion
I'm mad I spent 16 dollars on this	anger
I would recommend these to anyone looking to open up some closet space who is sick of ...	disgust
I wanted to watch something new and found this. I thought it would be scary but it was more of a thriller	fear
so happy that my niece got it on time christmas gift	happy
butt still hurts but not nearly as bad. cushion is fine just didn't solve problem. still glad I bought it	sad
I was amazed by the ending and I hope they there was a third book in the series it's amazing	surprise

Table 5.2: Some example reviews with corresponding emotion labels obtained via distant supervision in EWE.

- July 2014. Specifically, we use the aggressively deduplicated dataset which contains 82.83 million product reviews. Each review (considered as a document) is preprocessed by converting it to lowercase, tokenizing it with the NLTK toolkit[4] (punctuation is preserved as tokens), and filtering out reviews that are too short (less than 5 words). Note that, as the proposed weak labeling is not dependent on any domain-specific indicators of affect such as emoticons or hashtags, it can be easily generalized to any type of text documents. Table 5.2 presents a few examples of pre-processed reviews along with noisy emotion labels obtained following EWE.

[4]http://www.nltk.org/

5.3 Experiments

We evaluate the effectiveness of the proposed emotion embeddings in two setups: extrinsic evaluation on emotion classification (§5.3.1) and intrinsic evaluation on an emotion similarity task (§5.3.2).

5.3.1 Emotion Classification

The first task validates the effectiveness of the emotion embeddings under the supervised framework of emotion classification, where the learned word vectors are fed as features into classification models for predicting the emotion labels.

We train two classifiers: (i) L2-regularized multi-class logistic regression (LR) and (ii) support vector machine (SVM) based on LIBSVM (Chang and Lin, 2011), to predict the fine-grained emotion label at the sentence level. The results of 10-fold cross validation are reported in terms of macro F-score, which is the average F-score over all the emotion classes. F-score is the harmonic mean of precision (p) and recall (r), $\mathrm{F} = 2\frac{p \cdot r}{p+r}$.

5.3.1.1 Evaluation Datasets

Four benchmark emotion datasets (Alm, Aman, ISEAR and EmoTweet-28) originating from various genres of text are considered for emotion classification. The statistics of the datasets are summarized in Table 5.3. The datasets Alm, Aman and ISEAR were described earlier in section 4.3.1.

EmoTweet-28: While the other annotated datasets are modeled after existing emotion taxonomies, this corpus was created by inductively identifying a set of emo-

Dataset	Domain	# emotions	Total
Alm	fairy tales	5	1207
Aman	blogs	6	1290
ISEAR	experiences	5	5412
EmoTweet-top8	tweets	8	4664

Table 5.3: Statistics of emotion datasets

tion categories that characterize the emotions expressed in tweets (Liew et al., 2016). For our experiments, we extract a subset (**EmoTweet-top8**) of the original dataset comprising the eight most frequent emotions in the dataset, which include *amusement*, *anger*, *excitement*, *gratitude*, *happiness*, *hope*, *love* and *sadness*. Note that, emotion categories such as *gratitude* and *hope* do not currently feature in any emotion lexicon. Therefore, approaches relying solely on existing resources will not be able to classify such new and evolving taxonomies of emotions adequately.

5.3.1.2 Lexicons versus Representations

As the quality of the emotion embeddings depends on the underlying emotion lexicons adopted to create the training data to a notable extent, we analyze the results obtained using the source emotion lexicons directly versus using them to initialize EWE in Table 5.4.

For the emotion lexicons, we generate a feature vector consisting of the total number of words in the sentence associated with each emotion category. In essence, this baseline explores the use of emotion lexicons directly rather than using them to initialize EWE. For the word embedding models, since the input sentences can have

Methods		Alm	Aman	ISEAR
Lexicons	WNA	0.459	0.405	0.384
	NRC	0.387	0.370	0.378
	WNA+NRC	0.521	0.474	0.465
EWE_{UNI}	WNA	0.635	0.604	0.674
	NRC	0.604	0.582	0.666
	WNA+NRC	**0.661**	**0.623**	**0.679**
EWE_{MULTI}	NRC	0.630	0.602	0.666

Table 5.4: Comparison of using lexicons directly versus using lexicons to guide representation learning.

varying lengths, we compute the average of the word vectors of all the words in the sentence along each dimension to obtain the sentence representation as the input to the classification algorithm.

First, comparing $EWE_{UNI(NRC)}$ and $EWE_{MULTI(NRC)}$, it is observed that the multi-label model is much better than the single-label model. Moreover, we observe that the configurations using both the lexicons (WNA+NRC) yield better results than using any one lexicon alone. Lastly, all the EWE embeddings demonstrate significant improvements over using the lexicons directly, indicating that the affective word representation model learns useful information in addition to the knowledge available in the base lexicons adopted to guide the learning process.

Embeddings	Training corpus	Corpus size	$\lvert\mathcal{V}\rvert$
GloVe 6B	Wiki + Gigaword	6B tokens	400K
GloVe 42B	Common Crawl	42B tokens	1.9M
word2vec	Google news	100B tokens	3M
SSWE	tweets	10M tweets	137K
DeepMoji	tweets	1B tweets	50K
EWE	Amazon reviews	200K reviews	183K

Table 5.5: Details of compared embeddings.

5.3.1.3 Comparison Against State-of-the-art Representations

Next, we analyze the performance of EWE against state-of-the-art *generic embeddings* and *task-specific affective embeddings* described below, and summarized in Table 5.5. In particular, the following baselines are considered:

- ***Generic Embeddings***:
  - **GloVe**: Global vectors[5] for word representations (Pennington et al., 2014) trained on aggregated global word-word co-occurrence statistics from a corpus capture linear substructures of the word vector space. We use the vectors that were trained on: **GloVe 6B**: 6 billion words, uncased, from Wikipedia 2014 and Gigaword v5, of dimension $n = 300$; **GloVe 42B**: 42 billion words, uncased, from Common Crawl, of dimension $n = 300$.

[5]http://www-nlp.stanford.edu/projects/ glove/

- `word2vec`: These word representations[6] were learned with a continuous bag-of-words model (CBOW) (Mikolov et al., 2013a), where a target word is predicted given its surrounding context words. We use the vectors trained on 100 billion words of Google news dataset and are of $n = 300$.

- ***Affective Embeddings***:

  - **Sentiment-specific word embeddings (SSWE)**: These embeddings, obtained using a corpus of 10 million tweets, encode the sentiment information (derived using a set of positive and negative emoticons) of the text in the continuous representation of words[7] (Tang et al., 2014). We use embeddings that were trained with the unified model (SSWE_u).

  - **DeepMoji**: These word representations were obtained from a corpus of almost 1 billion tweets weakly labeled using a set of 64 emojis (Felbo et al., 2017).

  - **Retrofitting**: Instead of directly training task-specific affective embeddings such as SSWE and DeepMoji, Retrofitting (Faruqui et al., 2015) is a post-processing technique of tuning existing embeddings according to a task-specific lexicon. Using WNA as the source emotion lexicon, we apply Retrofitting to the generic word vectors (GloVe and `word2vec`).

The results of the emotion classification are presented in Table 5.6, with the generic embeddings model in the top half and affective embeddings in the bottom half. In general, we observe that GloVe 42B yields the second best results overall, and in line

[6] https://code.google.com/p/word2vec

[7] http://ir.hit.edu.cn/~dytang/paper/sswe/embedding-results.zip

Methods		Alm		Aman		ISEAR		EmoTweet-top8	
		LR	SVM	LR	SVM	LR	SVM	LR	SVM
GloVe 6B	d =300	0.548	0.583	0.547	0.555	0.648	0.643	0.574	0.581
GloVe 42B	d =300	<u>0.590</u>	<u>0.624</u>	<u>0.564</u>	<u>0.609</u>	<u>0.675</u>	<u>0.671</u>	<u>0.609</u>	<u>0.614</u>
`word2vec`	CBOW	0.413	0.433	0.424	0.478	0.655	0.661	0.526	0.568
SSWE	u	0.368	0.371	0.363	0.363	0.495	0.505	0.443	0.444
DeepMoji	d =256	0.300	0.275	0.332	0.336	0.598	0.607	0.533	0.560
Retrofitting	GloVe 42B	0.141	0.110	0.111	0.111	0.553	0.559	0.245	0.220
	`word2vec`	0.110	0.108	0.100	0.098	0.488	0.472	0.232	0.214
EWE_{UNI}	WNA+NRC	**0.632**	**0.661**	**0.602**	**0.623**	**0.679**	**0.679**	**0.610**	**0.618**

Table 5.6: Comparison against state-of-the-art word representations (*generic embeddings* in the top half; *affective embeddings* in the bottom half) on emotion classification. The best results are shown in **bold**, and the second best results are <u>underlined</u>. Paired t-tests using the results on all four datasets indicate EWE is significantly better than all the other methods with p-values < 0.02.

with other recent studies (Pool and Nissim, 2016), Retroffiting did not improve over any original word embeddings suggesting that post-processing word embeddings with respect to emotion knowledge requires additional considerations.

Additionally, the underlying model used to learn the word embeddings seems to be a crucial factor, as indicated by the results of GloVe 6B, GloVe 42B and `word2vec`. Although the GloVe 6B word representations were trained on 6 billion tokens, and GloVe 42B used a text corpus of size 42 billion tokens, they both considerably out-

perform the `word2vec` model which used a much larger corpus of 100 billion tokens.

Next, although SSWE and DeepMoji were both trained on tweets data, they perform very differently to each other, most likely due to their extremely different choices of affect spectrum (SSWE was modeled along binary polarities of sentiment, whereas DeepMoji used an axis of 64 categories), thus highlighting the importance of the emotion model adopted for creating the training dataset.

In addition, and rather surprisingly, all the generic embeddings (GloVe and word2vec) outperform all the affective embeddings (SSWE and DeepMoji) on all the four datasets. One possible reason for this could be due to the more generalizable sources of data that were used to induce the generic embeddings, while the affective embeddings were trained on tweets data, thus verifying the significance of the choice of the underlying text used to derive the representations.

Lastly, $\text{EWE}_{\text{UNI(WNA+NRC)}}$ statistically significantly outperforms all the other baselines across all the four datasets, confirming the effectiveness of the proposed method.

5.3.2 Emotion Similarity

The second task, a proof of concept qualitative analysis, compares the emotion similarity of the word vectors against the emotion similarity obtained from an emotion lexicon. In this experiment, the test affective information is derived from **DepecheMood (DM)**, an emotion lexicon consisting of 37,000 words and their emotion scores across eight affective dimensions summing up to 1 (Staiano and Guerini, 2014). DM was created using supervised training by applying distributional semantics to a dataset of crowd-annotated news articles. We consider the emotion label of a word

as the emotion category with the maximum affective weight. For example, the word "*serendipity*" which is represented as $(0.0, 0.014, 0.006, 0.005, 0.078, 0.230, 0.656, 0.006)$ corresponding to the emotion labels (*afraid, amused, angry, annoyed, dontcare, happy, inspired* and *sad*) in DM, is considered as being associated with the emotion *inspired*.

Following previous experimental setup for measuring affective consistency (Tang et al., 2014), we compute the accuracy of emotion similarity consistency between each emotion word and its top n nearest neighboring words as follows:

$$Accuracy = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\alpha\left(w_i, c_{ij}\right)}{m \times n} \tag{5.5}$$

where m is the number of words in the emotion lexicon, w_i is the ith word in the lexicon, c_{ij} is the jth closest word to w_i in terms of their cosine similarity, $\alpha\left(w_i, c_{ij}\right)$ is an indicator function, where $\alpha = 1$ if w_i and c_{ij} belong to the same emotion category and $\alpha = 0$ otherwise. The higher the accuracy, the better the clustering of emotionally similar words in the embedding space.

Table 5.7 presents the results of various embeddings, for $n = \{10, 20, 30\}$, where n is the number of nearest neighboring words. For fair comparison, for each word embeddings, only the words that appear in both the vocabularies (i.e., DM and word embeddings) have been used. Again, we observe that generic embeddings such as GloVe and `word2vec` outperform affective embeddings such as SSWE. However, the best results are obtained from EWE which have been specifically trained to capture some notion of emotion similarity in the form of emotion-enriched word representations.

Embedding	$n = 10$	$n = 20$	$n = 30$
SSWE_u	32.6	28.8	28.2
`word2vec`	35.5	33.1	30.2
GloVe	35.1	32.5	30.4
$\text{EWE}_{\text{UNI(WNA+NRC)}}$	**36.7**	**33.2**	**31.3**

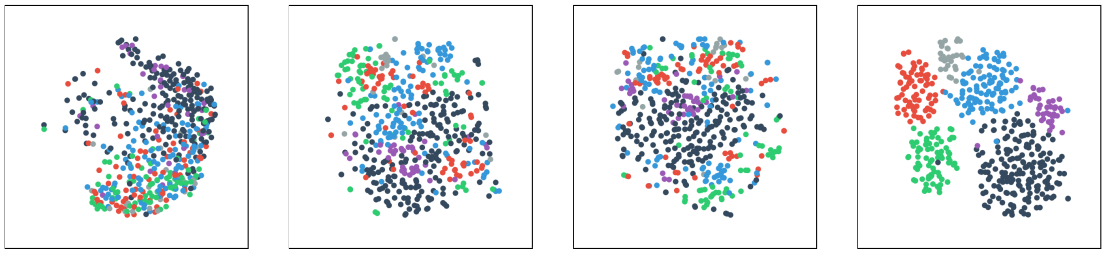
Table 5.7: Accuracy of emotion similarity tested on emotion lexicon DepecheMood

5.4 Qualitative Analysis

To further analyze the learned emotion embedding space, we use t-SNE (van der Maaten and Hinton, 2008) to visualize the word representations of a small subset of words obtained from SSWE (Tang et al., 2014), `word2vec` (Mikolov et al., 2013a), GloVe (Pennington et al., 2014) and EWE in Figure 5.3. The plots show that compared to other models, EWE is effective in clustering emotionally similar words into neighboring vector spaces.

One of the main objectives of learning emotion-enriched word representations is to capture the notion of emotion similarity. To illustrate this property, Table 5.8 presents the cosine similarity between a few emotionally similar ($\Uparrow$) and emotionally dissimilar ($\Downarrow$) word pairs, as obtained from various word embedding models.

Figure 5.4 shows confusion matrix plots providing an overview for some error analysis. In general, for imbalanced datasets such as Alm and Aman, it is observed that most misclassified instances are incorrectly labeled as *happy* class, likely because the *happy* class contains a disproportionately large number of training instances. Moreover, instances belonging to the *surprise* class are more often misclassified than



(i) SSWE (ii) `word2vec` (iii) GloVe (iv) EWE

Figure 5.3: t-SNE visualization of word embeddings for emotion words. The different colors represent the six different emotion categories.

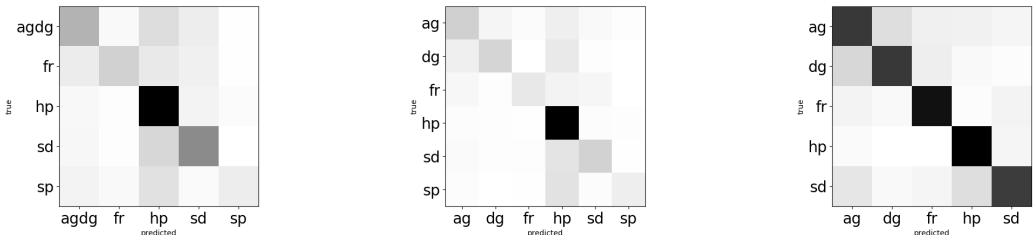
Word pairs	**GloVe**	**CBOW**	**EWE**
evil: tantrum ⇑, rejoice ⇓	0.13, 0.26	0.13, 0.25	0.27, 0.25
jolly: pleasing ⇑, dreadful ⇓	0.18, 0.20	0.33, 0.25	0.23, 0.12
hate: anger ⇑, funny ⇓	0.53, 0.53	0.36, 0.31	0.61, 0.43

Table 5.8: Cosine similarity between emotionally similar (⇑) and emotionally dissimilar (⇓) word pairs

correctly predicted, likely because the *surprise* class is highly underrepresented. Balancing the datasets might prove helpful. In ISEAR, *anger* and *disgust* classes are found to be confused with each other and *sadness* seems to be challenging.

5.5 Comparing EWE and SECO

Recall that previously in chapter 4, we presented a model called selective co-occurrences (SECO) for learning word-emotion association scores in an unsupervised manner from large text corpora. In this chapter, we incorporated a small amount of hu-



(i) Alm (ii) Aman (iii) ISEAR

Figure 5.4: Confusion matrix error analysis

man knowledge in the form of manually annotated emotion lexicons and leveraged a semi-supervised learning paradigm in order to further enhance models of emotion-enriched word representations (EWE). In short, SECO is considered a count-based measure of semantic similarity producing word-emotion association scores directly, whereas EWE is a neural-network model for generating word vectors which can be further used for computing word-emotion association through a metric such as cosine similarity. A detailed discussion of count-based models versus neural-network models was presented in chapter 3.

Earlier in this chapter (section 5.3.1.3), EWE was evaluated in a supervised setting where features derived from the generated emotion-enriched word vectors were input into supervised classification algorithms such as LR and SVM. However, word vectors can be used in both supervised and unsupervised emotion detection, a benefit over count-based methods such as SECO.

The next experiment, therefore, compares SECO and EWE under similar settings (i.e., unsupervised framework) to verify any additional benefits obtained through semi-supervised over purely unsupervised models. Table 5.9 presents the results of such a comparison, in terms of macro F-score. In this case, all text is pre-processed

Methods	Alm	Aman	ISEAR
SECO-PREC-NPMI$_{az}$	0.203	0.220	0.222
EWE$_{UNI}$	**0.376**	**0.288**	**0.236**

Table 5.9: Comparing the performances of EWE and SECO.

by converting to lowercase and tokenizing using the NLTK toolkit. The seed words "*angry*", "*disgust*", "*happy*", "*scared*", "*sad*" and "*surprise*" representing the six emotion categories *anger*, *disgust*, *happiness*, *fear*, *sadness* and *surprise*, respectively, are used to compute the word-emotion association scores.

From Table 5.9, we observe that EWE clearly outperforms SECO on all the three datasets, thus confirming the advantages received through the introduction of semi-supervised expertise gathered from existing emotion lexicons.

5.6 Conclusions

In this chapter, we have described a novel method of learning emotion-enriched word representations. The proposed approach leverages distant supervision to automatically obtain a large training dataset of text documents labeled with emotions and two neural network architectures for learning emotion embeddings. We demonstrated significant improvements over several baseline word representations in two tasks including emotion classification and emotion similarity. In addition, we presented a qualitative analysis visualizing the word vectors. Lastly, we also compared our two models, EWE (semi-supervised) and SECO (unsupervised), observing that additional benefits can be obtained through semi-supervised learning for emotion classification.

Chapter 6

Affective Representations for Sarcasm Detection

In this chapter, we shift our focus from word-level emotion association to the problem of sarcasm detection at document level. Sarcasm detection from text has gained increasing attention. While one thread of research has emphasized the importance of affective content in sarcasm detection, another avenue of research has explored the effectiveness of word representations. In this chapter, we introduce a novel model called Affective Word Embeddings for Sarcasm (AWES) for automated sarcasm detection in text. The proposed model learns affective word representations from weakly labeled data, which are then employed for building sarcasm classifiers. Specifically, first, a large corpus of reviews is automatically labeled with noisy affective labels. Then, a neural network model is trained to incorporate affective and contextual information into word representations. Finally, document-level representations derived from these affective word representations are used to classify sarcastic text. Extensive evaluation

on sarcasm detection on six datasets across three domains of text (tweets, reviews and discussion forum posts) demonstrates the effectiveness of the proposed model. The experimental results indicate that while sentiment affective representations yield best results on datasets comprising of short length text such as tweets, richer representations derived from fine-grained emotions are more suitable for detecting sarcasm from long length documents such as product reviews and discussion forum posts.

6.1 Introduction

Sarcasm is a sophisticated form of symbolic or nonliteral language use where one says or writes the opposite of what they mean. Due to this intentional ambiguity, detecting sarcasm, especially in written communication where the usual cues such as the tone of voice or facial expression are unavailable, is a particularly challenging task. Consider a few examples of sarcastic text utterances presented in Table 6.1 extracted from annotated sarcasm datasets.

Most existing sarcasm classification models have had to rely on a handful of small datasets as sarcasm is a remarkably rare positive class (Abercrombie and Hovy, 2016). While word representations have been used to overcome the issue of limited training data to some extent (Ghosh et al., 2015; Joshi et al., 2016b), most of these word embeddings have been obtained using only contextual information, without incorporating any affective information.

On the other hand, extensive research in psychology points towards a strong correlation between affect and sarcasm (Campbell and Katz, 2012; Filik et al., 2015; Larsen et al., 2016), a claim well supported by a number of recent models of computational

Domain	Text instance
tweet	*A little nervous to start school, 5 classes in one day should be fun...*
	I just thought of something that I could do.. but I'd need permission I guess..
review	*Guess what they said??? we dont replace phones with "physical damage"*
	Good luck getting this on once you've filled it, and good luck filling it

Table 6.1: Examples of sarcastic text

sarcasm detection incorporating affective information (González-Ibáñez et al., 2011; Joshi et al., 2015; Poria et al., 2016; Riloff et al., 2013; Sulis et al., 2016).

In light of these advancements, it seems reasonable that bridging the two avenues of research (i.e., word representations and affective knowledge) may bring additional benefits to sarcasm detection models. In the previous chapter, we introduced one such model, EWE, of learning affective word representations. While the model performed well in the task of emotion classification, it has two limitations:

(i) EWE followed a fine-grained model of emotions. However, as noted earlier, sarcasm detection models seem to benefit from both fine-grained (i.e., emotion) as well as coarse-grained (i.e., sentiment) spectrums of affect (discussed in detail earlier in chapter 3 section 3.2.3). Therefore, integrating sentiment-level affective knowledge may bring additional benefits.

(ii) EWE used an LSTM neural network architecture which reads data as an ordered sequence left to right. In other words, a given word receives information only about its left context. However, the right context of a word may also contain relevant

information. Therefore, capturing both the left as well as the right contexts of a word may further enhance the model.

Toward this end, in this chapter, we propose Affective Word Embeddings for Sarcasm (AWES), a framework for jointly modeling affective as well as contextual information, in order to obtain affectively richer word representations making them more suitable for detecting sarcasm in text. We investigate the use of information originating from two different spectrums of affect: sentiment and emotion. The proposed model projects words with similar affective orientations into neighboring regions of the embedding space. Specifically, a large corpus of product reviews is automatically labeled with noisy sentiment or emotion labels. Then, a Bidirectional Long Short-Term Memory (BLSTM) neural network model, which reads data left to right as well as right to left, is trained for incorporating the affective as well as contextual information into word representations, where the affective knowledge is derived via the noisy affective labels, and the sequences of words capture the contextual information.

The experimental results indicate that while the affective representations derived from a fine-grained model of emotions yield best results on datasets comprised of long text documents such as product reviews and discussion forum posts, affective representations modeling "simpler" sentiment information are more suitable for short text documents such as tweets.

The main contributions of this chapter can be summarized as follows:

- a framework for learning two types of novel affective word representations[1] (sentiment-aware and emotion-aware) for sarcasm detection using Bidirectional LSTM with automatically labeled training data;

[1] Available upon request.

- an extensive evaluation on six benchmark sarcasm datasets across three domains (tweets, product reviews and discussion forum posts), with the proposed model outperforming several relevant baselines;
- a novel finding that sentiment-aware word representations are most effective for short text sarcasm detection and emotion-aware word representations are most effective for detecting sarcasm on long texts.

The rest of the chapter is organized as follows. Section 6.2 presents the details of our proposed approach. The experiments and discussions of the results follow in Section 6.3. The model analysis are provided in Section 6.4 and finally, Section 6.5 concludes the chapter.

6.2 Affective Representations for Sarcasm Detection

In this section, we describe the details of our proposed framework for learning affective word embeddings for sarcasm. First, we present the details of the Bidirectional LSTM neural network model adopted for inducing affective word representations (§6.2.1). Then, we describe the process of automatically creating a large scale training dataset annotated with noisy affect labels (§6.2.2) and obtaining a document-level representation for detecting sarcasm (§6.2.3).

6.2.1 Learning Affective Representations

Recurrent neural networks such as Long Short-Term Memory (LSTM) can be effectively leveraged to obtain word representations, which are essentially the weights of the connections between the input and the hidden layers. A traditional LSTM model only captures the left context of a word. In other words, when predicting a target x_t, the LSTM model only considers the past sequence, $x_1, x_2, ..., x_{t-1}$. However, in textual data such as documents or sentences, the entire sequence is known beforehand and both the left and right contexts of surrounding words can contain useful contextual information. Therefore, we consider a Bidirectional LSTM (BLSTM) model (Graves, 2013; Hochreiter and Schmidhuber, 1997), which captures the *context* information (left-to-right as well as right-to-left sequence of words) by modeling the long-range dependencies between the words of a text document. Specifically, in a BLSTM, two separate LSTM's are trained, where the forward LSTM starts the recursion from x_1 and goes forwards, while the backwards model starts at x_T and goes backwards. The predictions from the forward and backward networks are then combined and normalized in an activation layer (e.g., a softmax layer). Additionally, we incorporate *affect* information into the word representations provided through the target label to map each word into an affective embedding space.

Figure 6.1 depicts an overview of the proposed framework consisting of a distant supervision module and an affective representation module. The distant supervision process is discussed later in section 6.2.2. In the affective representation module, first, all the words of the input text document are converted to their vector representation using an embedding matrix, which is sequentially fed (left-to-right and right-to-left)

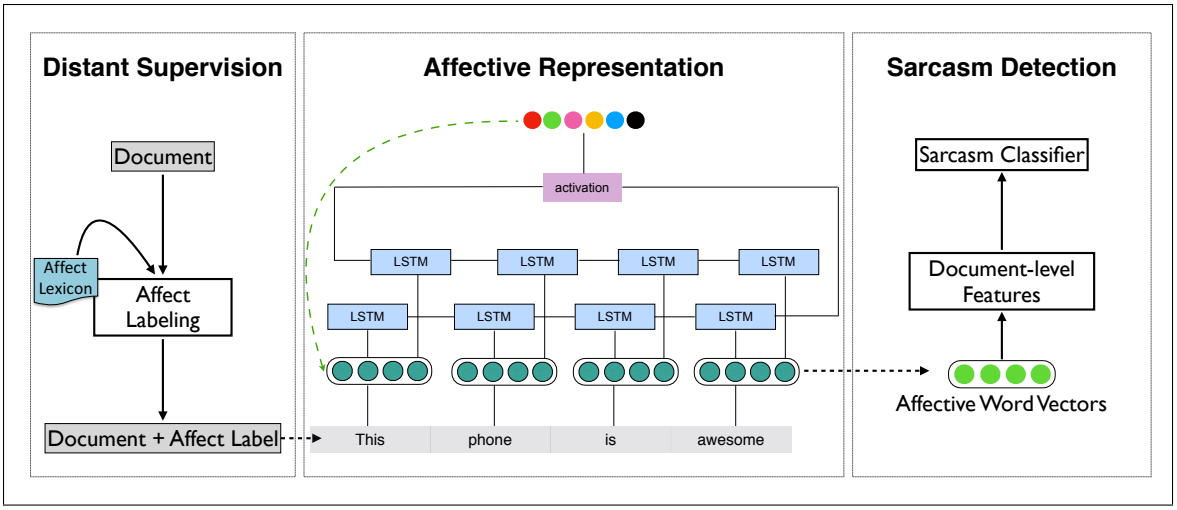


Figure 6.1: Overview of the proposed framework AWES.

to the BLSTM model. The outputs of the BLSTM are then flattened and connected to the output layer which is used to predict the target label. The objective function minimizes the loss between the predicted and true labels, and the training error derivatives are backpropagated to the embeddings in the first layer during the training process.

Given a document $d = \{w_1, w_2, ..., w_{|d|}\}$, where w_i denotes a word drawn from some vocabulary $\mathcal{V}$, the goal is to learn an affective word representation $e_i \in \mathbb{R}^n$ for $w_i \in \mathcal{V}$. The full embedding matrix is represented as $E \in \mathbb{R}^{n \times V}$, where n denotes the dimensionality of the embedding space and V is the size of the vocabulary. Typically, E can be initialized either randomly or through some pre-trained word embeddings.

We introduce two model architectures of AWES: one for capturing the sentiment information along binary dimensions such as positive and negative (AWES-senti), and the other for encoding a richer spectrum of emotions (AWES-emo).

6.2.1.1 AWES-senti

Essentially, AWES-senti follows the single-class setting, where an instance belongs to one of two classes (positive and negative). In other words, the target label y is binary represented, where $y = \{0, 1\}$. To predict the sentiment label of the input document, an output layer with a sigmoid activation function, which squashes the inputs into a probability range of $[0, 1]$ for every class, is added to the last layer.

The loss objective over a batch of m documents is calculated via binomial cross-entropy, minimized as follows:

$$\mathcal{L} = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log\left(y'_i\right) + \left(1 - y_i\right)\log\left(1 - y'_i\right)\right] \tag{6.1}$$

where i denotes the ith training sample, y is the binary representation of true sentiment label, and y' is the predicted probability.

6.2.1.2 AWES-emo

Assuming an emotion model of k classes, AWES-emo considers the multi-class setting where an instance can belong to one of the k emotion classes. Given an annotated document with its associated emotion label, the target value y is represented as a one-hot vector, where the values of all the indices but one are 0. For example, if a document d_i is labeled with emotion l_i, then:

$$y_j = \begin{cases} 1, & \text{if } y_j = l_i \\ 0, & \text{otherwise} \end{cases} \tag{6.2}$$

To predict the emotion label of the input document, an output layer with a softmax activation function which gives a probability distribution over the k classes is added on

top of the hidden layer for modeling multi-class probabilities. The softmax function converts the classification result into label probabilities, i.e. $y'_t \in [0,1]^k$.

The final training objective is to minimize the multinomial cross-entropy loss of the predicted and the true label distributions in order to fit the multi-class emotion labels, where the error is calculated as follows:

$$\mathcal{L} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k} y_{ij} \log\left(y'_{ij}\right) \tag{6.3}$$

where i denotes the ith training sample, j denotes the jth class, y is the true distribution (one-hot representation of the emotion label), and y' is the predicted probability distribution, $y'_{ij} \in [0,1]$ and $\sum_j y'_{ij} = 1$.

6.2.2 Weakly Labeled Data

In order to learn affective word embeddings, we require a large corpus of training instances along with their corresponding affective labels. Apart from the usual challenges involved in creating large scale manually annotated affective datasets (Mohammad and Turney, 2013), sarcasm being a rare positive class adds further complexity to the task (Abercrombie and Hovy, 2016). This difficulty is reflected in the sizes of current manually annotated sarcasm datasets, which range only around a few hundred instances each (see Table 6.5 for further details).

Distant supervision (Go et al., 2009) has been successfully leveraged in the past to automatically obtain large annotated datasets, albeit with noisy annotations. Previously, noisy labels have been derived by exploiting indicators of affect such as emoticons and hashtags (Mohammad, 2012), or emojis (Felbo et al., 2017), all of which are indicators specific for tweets data. This has led to prior work mostly focusing

on labeling data extracted from tweets (Felbo et al., 2017; Tang et al., 2014), whose vocabulary represents certain specific characteristics, in part due to tweets' restricted length of 140 characters, which forces users to get creative with their language. By contrast, we explore a novel, more generalizable source of data, a corpus of product reviews, where previously used affect indicators (such as emojis, emoticons, hashtags) are not applicable. Instead, we exploit an affect labeling algorithm to derive affective indicators from existing sentiment and emotion lexicons.

6.2.2.1 Data

Our data is extracted from a corpus of Amazon product reviews (McAuley et al., 2015), in particular, the aggressively deduplicated format containing around 80 million product reviews, spanning May 1996 to July 2014. All the reviews are tokenized using NLTK toolkit[2], which preserves punctuation as separate tokens, and reviews containing less than 5 tokens are filtered out.

6.2.2.2 Affect Labeling

Let $D = \{d_1, d_2, ..., d_{|D|}\}$ denote the set of unlabeled text documents. Given a document $d = \{w_1, w_2, ..., w_{|d|}\}$, $d_i \in D$, consisting of a sequence of words, the goal is to compute a corresponding affect label $l_i \in L$ for document d_i, where L denotes a pre-defined finite set of discrete affect labels. Primarily, there are two broad categories of affect: *sentiment*, consisting of binary labels such as positive and negative, and *emotion*, involving a more fine-grained spectrum of emotions such as *happiness*, *sadness*, *anger*, and so on. In this chapter, we seek to assess the effectiveness of both

[2]http://www.nltk.org/

of these models of affect. As a consequence, we create two sets of annotated datasets, D^{senti} and D^{emo}, following sentiment and emotion labeling, respectively. In the case of sentiment annotation, the set of labels $L^{sent} = \{positive, negative\}$, $|L| = 2$, whereas, typically, emotion annotation assumes $|L| > 2$.

First, for each word $w_i \in d$, an affect vector $\mathbf{a}(w) = \langle a_1, a_2, ..., a_{|L|} \rangle$ is computed, where a_j indicates the degree of word-affect association. The affect vector of a document $\mathbf{a}(d)$ is then, the sum of the affect vectors of all its words, i.e., $\mathbf{a}(d) = \sum_i \mathbf{a}(w_i)$. If the document has an association with at least one affect, i.e., $\exists a_j \in \mathbf{a}(d) \, | a_j > 0$, then, $l = \text{argmax } \mathbf{a}(d)$, where $l \in L$. In other words, documents assigned zero affect score (i.e., neutral) are not considered. Finally, in case multiple emotion labels have the maximum value, we randomly select one emotion label from the set of labels with the maximum score.

Three affect resources are leveraged for deriving the affect knowledge and computing the word affect vector $\mathbf{a}(w)$.

(i) **EmoLex** (Mohammad and Turney, 2013): For each word and each affect category, EmoLex denotes a binary association, where 1 indicates an association and 0 denotes no association with one or more of ten affect categories (Plutchik's eight emotions (Plutchik, 2001), which is a superset of Ekman's six emotions (Ekman, 1992), as well as positive and negative sentiments). For a given word w, we extract its binary association scores with positive and negative sentiment for calculating the sentiment label, and scores corresponding to the six categories of Ekman's model for computing the emotion label.

(ii) **SentiWordNet** (SWN) (Baccianella et al., 2010): For each word, SWN contains its positive and negative sentiment score. Unlike EmoLex, which only marks binary

Reviews	Sentiment	Emotion
I ordered this lunch bag, it's three days late, and it's still not here! I'm so angry. I want a refund.	negative	anger
The case is ok except for the slight smell coming from the case that I fear could be somewhat toxic.	negative	fear
The blue night light is beautiful. The led light gives a healthy glow and overall the Hood looks and works great	positive	happiness
The labels says plantronics but performance sucks so bad as if it's built in 1980's. It's dead on arrival. We can't hear anything.	negative	sadness
This system was better than expected. Once you get the hang of it it sharpens razor sharp. Good buy for most outdoorsman	positive	surprise

Table 6.2: Examples of reviews annotated with sentiment and emotion labels through distant supervision in AWES.

association, the strength of the association in SWN is specificed between the range of 0 and 1. For a given word w, we extract its corresponding sentiment association. (iii) **WordNetAffect** (WNA) (Strapparava and Valitutti, 2004): For each of Ekman's six emotions emotion, WNA specifies a list of words associated with that emotion. Here $a_j = 1$ if w is associated with the corresponding emotion, and $a_j = 0$ otherwise.

For D^{senti}, the sentiment labels are computed using EmoLex and SWN, while, for D^{emo}, the emotion labels are derived from EmoLex and WNA. In order to mitigate some noise in noisy labels, we constrain the labeling criteria such that only

Positive	**Negative**	**Total**
50.14%	10.97%	61.12%

ag	***dg***	***fr***	***hp***	***sd***	***sp***	**Total**
0.4%	0.04%	0.4%	32.9%	1.4%	1.1%	36.5%

Table 6.3: Distribution of affect labels for sentiment (*top*) and emotion (*bottom*). The emotion mapping is as follows: *ag*: anger, *dg*: disgust, *fr*: fear, *hp*: happiness, *sd*: sadness, *sp*: surprise

those documents where the labels output by both the lexicons in each case agree, are considered. Crucially, this motivates us to adopt the well-represented Ekman's model of six emotions (Ekman, 1992) rather than Plutchik's model of eight emotions (Plutchik, 2001), as to our knowledge, currently there does not exist a lexicon other than EmoLex containing annotations following Plutchik. Similarly, existing dimensional lexicons are all based on different emotion models, making them unsuitable for our labeling algorithm.

Table 6.2 illustrates a few sample reviews and their corresponding noisy affect labels and Table 6.3 presents the distribution of affect labels obtained through the process of distant supervision. While at least 60% of the original unannotated reviews were labeled with some sentiment label, only around 36% of the reviews are found to have any emotion label. This illustrates the difficulty in obtaining affect labeled datasets, even in case of noisy labels via distant supervision. Moreover, within the sentiment or emotion spectrums, there appears to be a huge variation in distribution, partly indicative of the nature of the underlying corpus of product reviews. Such

	Affect	# Reviews	V
D^{senti}	sentiment	400k	445k
D^{emo}	emotion	216k	183k

Table 6.4: Statistics of D^{senti} and D^{emo}

skew, nonetheless, is also observed in emotion annotated datasets. On one hand, there are almost five times more positive reviews than negative. On the other hand, the imbalance is further magnified in the emotion dataset, where almost 32% of the reviews are labeled with the *happiness* emotion, with all the other five emotions representing only about 1% or less, each.

Therefore, for each of the datasets, a balanced set of training instances is extracted, with an equal number of documents labeled with each affect. This is important as some affect categories are significantly underrepresented in our annotated dataset, which can cause the model to learn well from examples of some classes, but not the others. The final corpus statistics of D^{senti} and D^{emo} are summarized in Table 6.4.

Note that this approach of weak labeling documents with affect labels can be generalized to any type of text documents, as it is not dependent on any domain-specific indicators of affect such as emoticons or hashtags, which usually require careful curation and reasonable understanding of the affect representation in the source text.

6.2.3 Document Representation for Sarcasm Detection

The objective of our learning affective word embeddings is sarcasm detection, i.e., to classify a document as *sarcastic* or *non-sarcastic*. For such a purpose, we derive

a fixed size representation of a document by computing the element-wise minimum, maximum and average along each dimension of all the affective word vectors of all the words in the document.

Given a document $d = \{w_1, w_2, ..., w_m\}$ to be classified consisting of m words, where a word $w_j \in d$ is represented as a word vector $\mathbf{x}_j \in \mathbb{R}^n$ of dimension n, the goal is to compute a document-level representation $\mathbf{z}(d)$.

Let A denote a matrix, where the row vectors $I = 1, ..., n$ represent the word vectors, and the column vectors $J = 1, ..., m$ represent the words. Then, the document-level representation $\mathbf{z}(d)$ is obtained as follows:

$$\mathbf{z}(d) = [\, \mathbf{z}^{min}(d) \oplus \mathbf{z}^{max}(d) \oplus \mathbf{z}^{avg}(d)] \tag{6.4}$$

where $\mathbf{z}^{min}(d)$, $\mathbf{z}^{max}(d)$ and $\mathbf{z}^{avg}(d)$ denote the element-wise minimum, maximum and average, respectively, which are computed as follows:

$$\mathbf{z}^{min}(d)_i = \min_{j \in m} a_{ij}, \;\; i = 1, ..., n \tag{6.5}$$

$$\mathbf{z}^{max}(d)_i = \max_{j \in m} a_{ij}, \;\; i = 1, ..., n \tag{6.6}$$

$$\mathbf{z}^{avg}(d)_i = \frac{1}{m} \sum_{j \in m} a_{ij}, \;\; i = 1, ...n \tag{6.7}$$

where a_{ij} denotes the value of the element $A[i, j]$.

The intent is to capture the sentiment/emotion variations in a document. Given a set of labeled documents represented by such document vectors plus sarcasm labels, a supervised learning method can be used to learn a sarcasm detection model.

6.3 Experiments

6.3.1 Evaluation Datasets

The following six sarcasm datasets from three different domains of text are used for evaluating the proposed model. A summary of the statistics of the datasets is presented in Table 6.5.

SASI-AM and **SASI-TW**[3]: The SASI-Amazon (SASI-AM) dataset (Tsur et al., 2010) comprises of 180 sentences from Amazon product reviews annotated by three annotators as *sarcastic* or *non-sarcastic* on a scale of 1 to 5, where 1 means not sarcastic at all and 5 denotes clearly sarcastic. The scaling was then reduced to a binary classification where 1 and 2 were marked as non-sarcastic and 3, 4 and 5 as sarcastic instances. The inter-annotator agreement stood at Fleiss' $\kappa = 0.34$. Similar annotation scheme was followed for creating SASI-Twitter (SASI-TW), a dataset of 180 tweets (Davidov et al., 2010). This dataset's slightly better inter-annotator agreement (Fleiss' $\kappa = 0.41$) is attributed to the fact that in Twitter each tweet is generally self-sufficient and context-free, hence the sentiment in the sentence is expressed in a way that can be perceived more easily, whereas the sentences from Amazon product reviews were extracted from a full review, where the sarcasm could rely on other sentences.

RILOFF: This dataset contains sarcasm annotations for tweets (Riloff et al., 2013). As the authors share only the tweet IDs due to Twitter's data sharing restrictions, by the time we tried to download the tweets, we could obtain only a subset of the original dataset.

[3]http://people.seas.harvard.edu/~orentsur/data/sarcasmData.tar

Dataset	Domain	Sarcastic	Non-sarcastic	Total
SASI-TW	tweets	73	107	180
RILOFF	tweets	112	498	610
ELECT	tweets	938	938	1876
SASI-AM	reviews	67	113	180
FILATOVA	reviews	437	437	874
IAC-SARC	forum posts	1630	1630	3260

Table 6.5: Statistics of sarcasm evaluation datasets.

ELECT: This dataset includes crowdsourced annotations of tweets (Mohammad et al., 2013) pertaining to the 2012 US presidential elections, along dimensions such as emotions, style and purpose, where the style component of the annotations includes subcategories such as simple statement/question, exaggeration/hyperbole, sarcasm, rhetorical question, understatement, weird and humorous. We assume 938 tweets annotated as sarcasm to be *sarcastic*, and extract 938 simple statements as denoting *non-sarcastic* instances.

FILATOVA[4]: This consists of a corpus of reviews, marked as sarcastic or not, using crowdsourcing (Filatova, 2012). We evaluate on a balanced dataset, with all 437 sarcastic reviews and a random subset of 437 (of 817) regular reviews.

IAC-SARC: The sarcasm corpus v2 (Oraby et al., 2016) contains quote-reponse pairs from a dataset of discussion forum posts, where the quote functions as a dialogic parent to the response. As the sarcasm annotations relate only to the response text,

[4]http://storm.cis.fordham.edu/~filatova/SarcasmCorpus.html

we extract 1,630 responses per class (*sarcastic* and *non-sarcastic*) from the general sarcasm category.

In order to maintain consistency across all the different datasets and domains, all the text is tokenized using NLTK, all username references (e.g., @yorku) are replaced with a REF token and all URLs (e.g., "http://yorku.ca") are replaced with a LINK token.

6.3.2 Baselines

All the approaches are evaluated using 10-fold cross validation and the results are reported in terms of macro-averaged F-score score over the two classes, *sarcasm* and *non-sarcasm*, where F-score is the harmonic mean of precision and recall. The $l2$-regularized logistic regression model implemented in the scikit library is used for classification. In particular, the following baselines are considered:

- ***n*-grams**: n-grams are one of the most effective features leveraged in sarcasm detection (Liebrecht et al., 2013; Lukin and Walker, 2013). We implement baseline models exploiting n-grams features including *uni*grams (uni), *bi*grams (bi) and *tri*grams (tri), indicating the presence or absence of each n-grams.

- **Riloff** (Riloff et al., 2013): This baseline represents sarcasm as a contrast between positive and negative sentiment. We re-implement three of their rule-based algorithms: (i) Positive (pos): an instance is labeled as sarcastic if it contains any positive term; (ii) Negative (neg): an instance is labeled as sarcastic if it contains any negative term; and (iii) Positive and Negative (posneg): an instance is labeled as sarcastic if it contains both a positive and a negative

sentiment term, in any order. In this case, the sentiment scores are derived from EmoLex (Mohammad and Turney, 2013).

- **Joshi** (Joshi et al., 2016b): This baseline models sarcasm as a discordance between semantic similarity, obtained via word embeddings. We re-implement their baseline consisting of unigrams, bigrams and trigrams features (Liebrecht et al., 2013) augmented with similarity features computed from `word2vec` word vectors (Mikolov et al., 2013a).

- **Word vectors**: Another relevant baseline compares contextual word vectors including GloVe (Pennington et al., 2014) and `word2vec` (Mikolov et al., 2013a), as well as affective word vectors including Sentiment Specific Word Embeddings (SSWE) (Tang et al., 2014) and DeepMoji (Felbo et al., 2017). The specifications of the word vectors are as follows: **GloVe** trained on a corpus of 42 billion words from Common Crawl, of dimension $d = 300$; `word2vec` continuous-bag-of-words model trained on 100 billion words of Google news dataset, $d = 300$; **SSWE** unified model trained on a corpus of 10 million tweets, $d = 50$; **DeepMoji** pretrained vectors from 1.5 billion tweets, $d = 256$.

6.3.3 Results

The main experimental results summarized in Table 6.6 facilitate a few general observations such as n-grams methods performing better on long text documents than on short texts, as expected; the Riloff et al. (2013) approach of detecting contrast between positive and negative sentiments works better for short text documents; similarly to n-grams, all the word vectors methods also perform better on long texts

method		short text			long text			average		
		SasiTW	Riloff	Elect	SasiAM	Fila	IAC	short	long	all
n-grams	uni	0.53	0.60	0.60	0.58	0.69	0.62	0.58	0.63	0.60
	uni+bi	0.58	0.61	0.60	0.56	0.69	0.62	0.60	0.62	0.61
	uni+bi+tri	0.58	0.54	0.59	0.50	0.67	0.62	0.57	0.60	0.58
Riloff	pos	0.57	0.51	0.45	0.51	0.35	0.40	0.51	0.42	0.47
	neg	0.56	0.53	0.53	0.51	0.46	0.45	0.54	0.47	0.51
	posneg	0.54	0.63	0.48	0.50	0.46	0.43	0.55	0.46	0.51
Joshi	-	0.58	0.57	0.61	0.52	0.67	0.64	0.59	0.61	0.60
word vectors	GloVe	0.54	0.75	0.60	0.60	0.72	0.69	0.63	0.67	0.65
	`word2vec`	0.52	0.73	0.60	0.59	0.72	0.70	0.62	0.67	0.64
	SSWE	0.59	0.63	0.59	0.57	0.66	0.67	0.60	0.63	0.62
	DeepMoji	0.53	0.64	0.60	0.65	0.71	0.71	0.59	**0.69**	0.64
AWES	senti	0.57	0.76	0.62	0.61	0.74	0.70	**0.65**	0.68	**0.67**
	emo	0.55	0.76	0.61	0.64	0.74	0.70	0.64	**0.69**	**0.67**

Table 6.6: Results (macro F-score) of sarcasm detection across six datasets. The last three columns contain the average result of each method on short text, long text and all text, respectively.

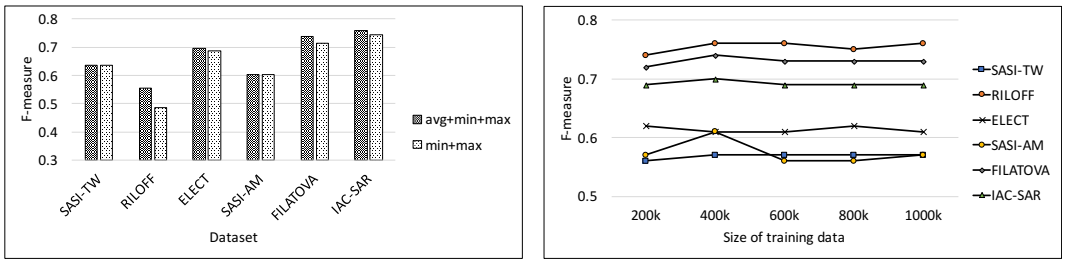
than on short texts; and, on average across all the six datasets, our proposed model (AWES) outperforms all the other baselines. Paired t-tests using the results on all six datasets indicate AWES-senti or AWES-emo is significantly better than all other methods except DeepMoji with p-values < 0.05.

Comparing the performance of n-grams to word embeddings-based features, although the n-grams models set a competitive baseline, in general, the word embeddings features yield additional improvement, suggesting that depending on the type

of features derived from word embeddings, word embeddings-based features alone can be useful for sarcasm detection. Note that this finding is in contrast with the results reported in an earlier study (Joshi et al., 2016b). We believe this may be because the features employed by Joshi et al. (2016b) consist of high-level similarity scores derived from word embeddings, whereas our features (element-wise minimum, maximum and average word vectors) are low-level features, directly modeling the word vectors.

In order to assess the effectiveness of our proposed approach, we compare the results of AWES against those of four pre-trained word embeddings (GloVe, word2vec, SSWE and DeepMoji). On average, AWES-senti achieves the overall best result on short text documents, while AWES-emo (along with DeepMoji) obtains the best result on long text documents. However, we observe that, in general, the performance of DeepMoji word vectors is worse than all the other word embeddings on short text documents, and while SSWE obtains the highest score on the SASI-TW dataset, it falls short on all the remaining five datasets. On the other hand, AWES performs consistently well on *both* the short and long text domains.

One of the most interesting observations of this study, however, is as follows: while both DeepMoji and SSWE were trained on short text data (tweets), their performance is strikingly different. DeepMoji, trained on a set of emojis performs better on long text documents, whereas, SSWE, trained on binary sentiment categories is better on short texts. This observation is in line with the performance of AWES, where AWES-senti obtains the best average score on short texts, while AWES-emo achieves the best average score on long texts. In summary, the results suggest that the choice of the affect model (i.e., sentiment versus emotion) used to train the word vectors is a more distinctive factor than the domain of training data (i.e., tweets such as in the



(a) (b)

Figure 6.2: Model analyses: (a) effect of document-level features, and (b) effect of size of training data.

case of SSWE and DeepMoji versus product reviews as used for AWES), leading us to conclude, albeit counterintuitively to a certain extent, that richer models of emotions do not always lead to additional gains, and that "simpler" models of affect along the axes of positive and negative sentiment are actually better for short text domains.

6.4 Model Analysis

6.4.1 Effect of Document-level Features

To assess the individual effect of the document-level features derived from word vectors, we plot the F-score results obtained on each dataset with all the three features (minimum, maximum and average of word vectors along each dimension) against using only the features involving the minimum and maximum. The plots in Figure 6.2(a) highlight the importance of using all the three features.

6.4.2 Effect of Size of Training Data

Figure 6.2(b) presents the results of using different sizes of training data. We observe that the maximum result is achieved with around 400k reviews, and remains consistent thereafter. This is beneficial in the sense that, unlike other models (Felbo et al., 2017; Tang et al., 2014) which require millions of training instances, our model is able to learn effective representations from a relatively "small" number of training instances, thereby requiring considerably less training time.

6.5 Conclusions

In this chapter, we introduced a novel method for detecting sarcasm in text by leveraging affective word representations obtained from weakly labeled data. Extensive evaluation on six sarcasm datasets of short and long text documents demonstrates the effectiveness of the proposed framework. In particular, it was observed that sentiment affective word representations are more suitable for short text documents such as tweets, whereas emotion word representations benefit sarcasm detection in long documents such as product reviews and discussion posts.

Chapter 7

Leveraging Transitions of Emotions for Sarcasm Detection

One popular thread of research in document-level computational sarcasm detection involves modeling sarcasm as a contrast between positive and negative sentiment polarities or exploring more fine-grained categories of emotions such as *happiness*, *sadness*, *surprise*, and so on. Most current models, however, treat these affective features in a discrete manner, without any regard for the continuous information contained between the affective states. In order to explore the role of transitions in affective states, in this chapter, we formulate the task of sarcasm detection as a sequence classification problem by leveraging the natural shifts in various emotions over the course of a piece of text. Experiments conducted on two evaluation datasets demonstrate the potential of employing emotion transitions for sarcasm detection, with the proposed model outperforming several baseline models.

7.1 Introduction

The strong correlation between affective states and sarcasm has been well-highlighted by various research studies in psychology (Boylan and Katz, 2013; Campbell and Katz, 2012; Colston, 1997; Filik et al., 2015, 2016; Jorgensen, 1996; Kreuz et al., 1991; Larsen et al., 2016; Phillips et al., 2015) as well as well-supported by recent works in computational sarcasm detection (González-Ibáñez et al., 2011; Poria et al., 2016; Sulis et al., 2016).

In addition to the commonly used lexical features such as n-grams, punctuation, number of words, etc., some computational models of sarcasm detection also employ discrete affective features such as the number of positive words, number of negative words, number of emotion words, and so on. To further increase the vocabulary coverage, some models also leverage word embedding features. In the previous chapter, we presented one such model, AWES, for document-level sarcasm detection through affective word embeddings.

A severe limitation of the current methods, however, is that they all employ *discrete* features (e.g., minimum, average, sum, binary representation, etc.) in designing their models. In other words, the role of *continuous* information in the form of affective transitions within sarcastic instances remains unexplored as yet. We hypothesize that additional advantages may be acquired by distinguishing between discrete and continuous information, where the latter takes into considerations the sequences or transitions of affective states when designing sarcasm detection systems.

Given an instance of text, the task at hand entails labeling it as *sarcastic* or *non-sarcastic*. Consider an example of a sarcastic tweet, "*Woke up and now having a*

headache. Great!" While the first part of this tweet could be considered as conveying a *sad* or an *angry* tone, the over-enthusiastic "*Great!*" at the end of the tweet exudes a hint of sarcasm. Our goal is to investigate whether such sequential variance in emotions (e.g., from *sad/angry* to *happy*) could be leveraged effectively for detecting sarcastic instances.

In this chapter, we seek to investigate the three following research questions:

- RQ1: Are finer-grained categories of emotions more useful than binary sentiment polarities for computational sarcasm detection?
- RQ2: Do sequences of emotion *transitions* provide more discriminative value than discrete emotion features?
- RQ3: Finally, which emotion transitions in particular are better suited for building sarcasm classifiers?

To address the challenging task of computational sarcasm detection, we formulate it as a sequence classification problem in order to design our proposed model, which we call Emotion Transitions for Sarcasm (ETS). Each instance of text such as a document or sentence is first divided into a number of consecutive non-overlapping chunks. Then, for each chunk, a vector of emotion features is computed by employing various emotion resources. Finally, the emotion vectors are fed into sequence classification models to learn patterns of sarcasm. Experiments on two sarcasm datasets (Amazon product reviews and Twitter tweets) demonstrate the potential of using emotion transitions, with the proposed model outperforming several baseline models.

The main contributions of this chapter can be summarized as follows:

- To the best of our knowledge, we present the first analysis of distinctive *transitions* within emotion sequences in sarcasm versus non-sarcasm text;
- We describe a novel method, ETS, to leverage the inherent transitions of emotions within text for automatically detecting sarcasm using sequence classification;
- We demonstrate the effectiveness of the proposed approach by comparing against several baselines on two evaluation datasets;
- Under qualitative analysis, we conduct extensive experiments to investigate which emotion category or combination of emotion categories is most discriminative for sarcasm detection.

The rest of the chapter is organized as follows. Section 7.2 presents the details of our proposed approach. The experiments and discussions of the results follow in section 7.3. A qualitative analysis is provided in section 7.4, and a comparative experiment with a previously introduced model (AWES) is described in section 7.5. Finally, section 7.6 concludes the chapter.

7.2 Sarcasm Detection using Emotions

We formulate the task of sarcasm detection as a sequence classification problem in order to capture the sequential transitions in emotional information. Consider the input sentence s represented as a multidimensional sequence $\mathbf{X}$, where $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$ denotes an ordered list of T feature vectors. For example, $\mathbf{X}$ could be a sequence of words or phrases or emotion vectors. Let $y \in \{1, 0\}$ denote the binary class labels,

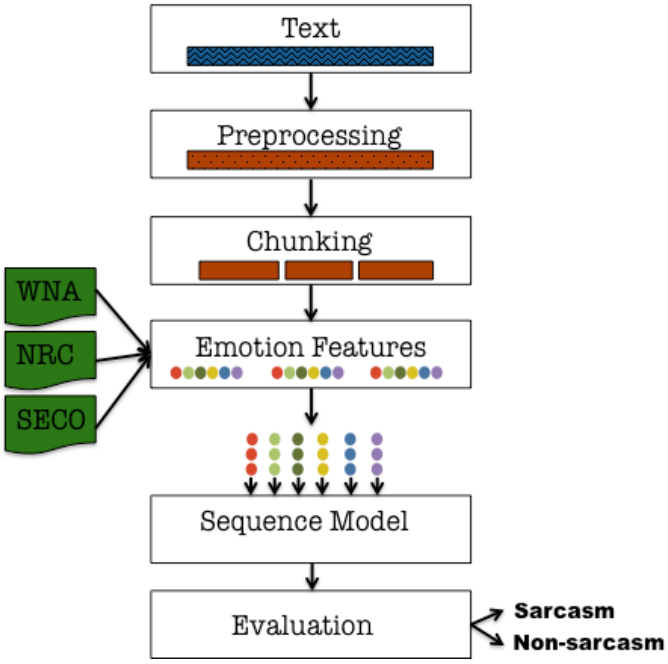


Figure 7.1: Overview of the proposed framework ETS.

where *sarcasm* ($y = 1$) indicates a sarcastic instance and *non-sarcasm* ($y = 0$) indicates a non-sarcastic instance. The goal is to predict the class label y given an input sequence $\mathbf{X}$.

As the central idea of our proposed approach relies on identifying the transitions between various emotions in text, first, we segment the sentence s into smaller chunks, then represent each chunk in the form of affective features and subsequently, utilize these features to train learning algorithms to detect sarcasm. The overview of the proposed framework is presented in Figure 7.1.

7.2.1 Obtaining Chunks

In order to build a model for capturing the transitions within emotions, first, the input sentence $s = \{w_1, w_2, ..., w_l\}$ of length l is decomposed into a set of n non-overlapping segments called chunks, $C = \{c_1, c_2, ..., c_n\}$, where $2 \leq n \leq l$ (as a chunk consists of one or more consecutive words and there are at least 2 chunks in a sequence). In addition, k denotes the length of chunk, i.e., the number of of words in c, and $1 \leq k < l$, where different chunks can have different number of words. For example, when $k_i = 1$, chunk c_i consists of one word, $k_i = 2$ means chunk c_i consists of two consecutive words and so on. Thus, as the size of k increases, the number of chunks n decreases.

We investigate three possible methods of obtaining such chunks: phrase-based, equal-n and fixed-k.

- **Phrase-based chunking**: Intuitively, a sentence can be decomposed into a sequence of phrases. We employ a shallow parser (Punyakanok and Roth, 2001) to identify non-overlapping and non-embedded syntactical phrases such as noun and verb phrases in natural language text. This approach yields a sequence composed of chunks consisting of variable number of words, and therefore, variable number of chunks.

- **Equal-n chunking**: Since the unconventional language typically expressed in tweets can be challenging for shallow parsing (as many other NLP tasks), we also explore the variation of dividing all the instances into the same number of chunks (equal-n).

- **Fixed-k chunking**: This variant explores having a fixed number of consecutive

words per chunk (fixed-k).

7.2.2 Computing Emotion Scores

Each chunk c_i is, then, represented in terms of an emotion vector $\mathbf{e}(c_i) = \langle e_1, e_2, ..., e_m \rangle$, where $e_j \in \{0, ..., 1\}$ indicates the real-valued degree of an emotion e_j from the set of emotions $E = \{e_1, e_2, ..., e_m\}$, normalized between the range of 0 and 1. We choose Ekman (1992) model of emotions consisting of $m = 6$ basic emotion categories, namely, *anger*, *disgust*, *fear*, *happiness*, *sadness* and *surprise*, as this model is well-represented in many emotion lexicons freely available for research. As a consequence, we utilize the following three resources for computing the vector of emotion scores.

- **WordNet Affect (WNA)**: One of the earliest emotion lexicons, WNA (Strapparava and Valitutti, 2004) was created by manually labeling about 1,314 synsets with one or more of Ekman's six emotions. For each emotion category, WNA specifies a list of words that are associated with that emotion. The emotion vector of a word w is represented as $\langle s_1, s_2, ..., s_6 \rangle$, where s_i is either 0 or 1, depending on whether w is associated with the emotion or not.

- **NRC EmoLex (EmoLex)**: Developed using crowd-sourcing, one of the largest manually annotated emotion lexicons to date, EmoLex (Mohammad and Turney, 2013) contains about 14,200 unigrams annotated with one or more of ten affect categories (Plutchik (2001) eight emotions as well as positive and negative sentiments). For instance, the binary association between the word *"awful"* and 10 affect categories (anger, anticipation, disgust, fear, joy, negative, positive, sadness, surprise, trust) is represented as: $1, 0, 1, 1, 0, 1, 0, 1, 0, 0$. For a

Emotion	Seed words
anger	*anger, angry, angrily, annoy, furious, irate, rage, violence, irritate, mad*
disgust	*disgust, horrified, ashamed, embarrass, dislike, distaste, hatred, hate, sick, stupid*
fear	*fear, afraid, worry, danger, threaten, frighten, horror, scare, terror, terrify*
happiness	*happy, happier, happily, love, glad, joy, delight, fun, smile, cheer*
sadness	*sad, sorrow, anguish, grief, despair, regret, terrible, sadden, death, cry*
surprise	*surprise, unexpected, stun, astonish, shock, excite, amaze, awe, wonder, exclaim*

Table 7.1: Seed words for word-emotion association using SECO.

given word w, we extract its binary association scores corresponding to the six categories of Ekman's model.

- **Selective Co-occurrence word-emotion association scores (SECO)**: The two previous resources involve fixed-sized manually created emotion lexicons. To obtain the affective information for words not in these lexicons, we leverage SECO (Agrawal and An, 2016), described earlier in chapter 4, to expand our coverage by automatically computing word-emotion association from a large text corpus such as Wikipedia. SECO takes as an input a list of seed words per emotion category and computes a vector of real-valued emotion scores for a word based on the semantic similarity between the word and the list of seed words for each emotion category. The number of seed words for each category is much smaller than the number of words in a lexicon, ranging normally from

1 to a few. Given a word w and an emotion category e, the word-emotion association score between w and e is computed by averaging the similarity scores between w and each of the seed words for e. Following SECO's recommended settings for counting selective co-occurrences, we use normalized Pointwise Mutual Information (NPMI) to measure the similarity between two words and set the position = *preceding* and window size = 15 (i.e., only the closest seed word *preceding* the word w within a context window of 15 words is considered as co-occurring with w). Additionally, we initialize the model using ten seed words per emotion category, listed in Table 7.1.

In order to illustrate the procedure of computing emotion scores, as an example, consider the following input document, *"I am almost never motivated to write a review but my set crapped out a few weeks after the warranty. Yay!*".

1. The first step involves preprocessing the sentence by removing the stopwords and retaining words from four parts-of-speech (i.e., nouns, verbs, adjectives and adverbs) by applying pos-tagging.

2. Then, consider it rendered into a set of $n = 6$ chunks using the phrase-based chunking approach, $C = \langle$ *"almost never motivated write", "review", "set crapped", "weeks", "warranty", "yay"*$\rangle$.

3. Next, for each word w, we compute its word-emotion association $\mathbf{e}(w)$ by using all the three emotion resources (WNA, EmoLex and SECO).

4. Then, the emotion vector of a chunk $c_i \in C$ composed of k_i words is obtained by computing the average of the emotion scores of all its words, $\mathbf{e}(c_i) =$

$\frac{1}{k_i} \sum_{j=0}^{k_i} \mathbf{e}(w_j)$.

5. Finally, the sequence of chunks C is represented in terms of concatenated emotion vectors, resulting in a sequence of feature vectors $\mathbf{X} = (\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_n)$.

7.2.3 Classifying Sequences for Sarcasm

To build a classifier using sequential patterns of multidimensional emotion data, we adopt the following three models: hidden Markov model (HMM), Long Short-Term Memory networks (LSTM) and classification of sequence-derived features using Naive Bayes (NB) and Support Vector Machine (SVM).

7.2.3.1 Hidden Markov Model (HMM)

A common model for sequence classification, hidden Markov model (HMM) models a joint distribution over states and observations (Rabiner and Juang, 1986). We adopt Gaussian HMM[1] to model our multivariate data of numeric sequences, which is an ordered sequence of observations, with each observation consisting of measurements from six numeric covariates (one representing each emotion). The number of observations corresponds to the number of chunks.

We consider the transitions between two neighboring states, from left to right, where the current state depends only on its previous state, i.e., the transitions follow a first order Markov process. HMM assumes a series of discrete time steps $t = 1, 2, ..., T$, where an item Y_t is observed at step t. At any given time, the HMM is in some state, chosen from a finite set of discrete states, $Q = \{q_o, q_1, ..., q_p\}$. At each time step t a

[1] http://doc.gold.ac.uk/~mas02mg/software/hmmweka/

latent state q_t is transitioned from the previous state q_{t-1} in a Markovian manner, i.e., $P(q_t \mid q_{t-1}, q_{t-2}, ..., q_1) \equiv P(q_t \mid q_{t-1})$, known as the transition probability. A full HMM is specified by a triplet of parameters including the distribution of the initial states, a transition matrix of the state transition probabilities and an emission matrix of the observation transition probabilities (Rabiner and Juang, 1986). The learning is solved using the forward-backward algorithm and Viterbi algorithm is used for decoding to calculate the most probable state sequence.

7.2.3.2 Long Short-Term Memory (LSTM)

Unlike HMM, neural networks can capture long-range dependencies within sequences, which might be useful for sarcasm detection. In this chapter, we adopt Long Short-Term Memory network (LSTM), a special kind of recurrent neural network widely used to model sequence data (Hochreiter and Schmidhuber, 1997). Our model (illustrated in Figure 7.2) is composed of six merged LSTM encoders for classification over six parallel sequences, one for each emotion. The six emotion sequences are encoded into vectors by the six separate recurrent LSTM modules, which are then concatenated. A fully connected network is then trained on top of this concatenated representation, followed by an output layer with sigmoid activation function for binary classification.

Given a multidimensional sequence $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$, where $\mathbf{x}_i$ is a vector of 6 scores, one for each emotion, and T is the number of chunks (i.e., time steps), the LSTM model processes it sequentially. For each position $\mathbf{x}_t$, given the previous output $\mathbf{h}_{t-1}$ and cell state $\mathbf{c}_{t-1}$, an LSTM cell uses the input gate $\mathbf{i}_t$, the forget gate $\mathbf{f}_t$ and the output gate $\mathbf{o}_t$ together to generate the next output $\mathbf{h}_t$ and cell state $\mathbf{c}_t$. The output

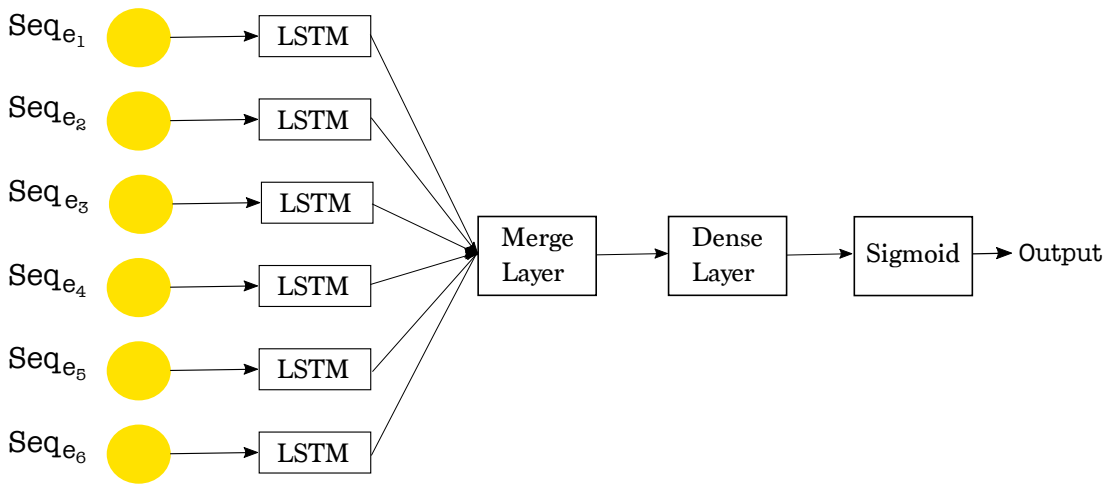


Figure 7.2: LSTM architecture for ETS. Seq e_i represents a sequence of emotion scores (one score for each chunk) for emotion e_i.

of an LSTM layer is a sequence of hidden vectors $[\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_T]$. Each annotation $\mathbf{h}_t$ contains information about the whole input sequence with a strong focus on the parts surrounding the t-th element of the input sequence. The final feature representation $\mathbf{h}$ is then used to predict the binary class labels (*sarcastic* or *non-sarcastic*) for unseen test sequences. We used binary cross entropy as the loss function for the two class classification, and Adam algorithm (Kingma and Ba, 2014) for optimization.

7.2.3.3 Classification with Sequence-Derived Features

While the two models described above (HMM and LSTM) can acccept sequence data directly, conventional classification methods such as Naive Bayes (NB), Support Vector Machine (SVM), etc., are designed for classifying feature vectors which can be obtained by transforming a sequence into a vector of features. Although the sequential nature of the sequence is lost during such a transformation, the following extracted

features may provide additional information beneficial for building a classifier.

We derive the following features from the numeric emotion vector sequence $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$ computed from each of the three emotion resources (WNA, EmoLex and SECO), where T is the length of $\mathbf{X}$, i.e., the number of chunks:

- **Basic Features (BF)**: six emotion scores per chunk for a total of $T \times 6$ features.
- **Extended Features (EF)**: For each chunk $\mathbf{x}_t$, the minimum emotion category (T features), the maximum emotion category (T features), the average emotion score (T features). For the entire sequence $\mathbf{X}$, the variance of all the chunks' average emotion scores (1 feature), the transition between each chunk's average emotion score (i.e., up/down/no_change) ($T - 1$ features), the average of each emotion (6 features), the variance of each emotion (6 features), the standard deviation of each emotion (6 features). The variance and standard deviation, in particular, are helpful for measuring the degree of emotion changes between the chunks.
- **Features + *n*-grams**: As n-grams have been shown to be quite effective features in sarcasm detection (Liebrecht et al., 2013; Lukin and Walker, 2013), we further add n-grams to our list of features.

7.3 Experiments

7.3.1 Evaluation Datasets

We employ two sarcasm datasets, SASI-Amazon and SASI-Twitter (described earlier in section 6.3.1) to evaluate the performance of our proposed approach. Unlike other

Dataset	Sarcasm	Non-sarcasm	Total
SASI-Amazon	67	113	180
SASI-Twitter	73	107	180

Table 7.2: Statistics of sarcasm datasets

sarcasm datasets, these two datasets were specifically chosen as they were manually annotated for sarcasm rather than automatically collected through distant supervision using hashtags. This is motivated in part by the observation that typically hashtag annotation is noisy and biased (Abercrombie and Hovy, 2016; Davidov et al., 2010; Riloff et al., 2013). Nonetheless, the low inter-annotator agreement scores of the two evaluation datasets (Fleiss' $\kappa = 0.34$ and Fleiss' $\kappa = 0.41$, respectively) highlight the inherent difficulty of deciphering sarcasm, even for humans (Abercrombie and Hovy, 2016; González-Ibáñez et al., 2011; Kreuz and Caucci, 2007). Table 7.2 summarizes the statistics of the evaluation datasets while Table 7.3 presents some sample instances.

7.3.2 Baselines

We compare our proposed approach against several baselines described below.

- ***n*-grams**: n-grams are one of the most effective features leveraged in sarcasm detection (Liebrecht et al., 2013; Lukin and Walker, 2013). We implement baseline models exploiting n-grams features (unigrams, bigrams and trigrams) using two popular text classification algorithms: NB and SVM.

Dataset	Instance	Label
$SASI_{am}$	*I love my Kindle, I have to leave my house and drive 10 minutes up the road to get it to work.*	sarcasm
	Good luck guessing where the GPS wants you to go	sarcasm
	The fifth book of the Harry Potter series brings a little bit disappointment to me.	non-sarcasm
$SASI_{tw}$	*Traded in my useless NEW phone today for another one. Technology is wonderful when it works*	sarcasm
	Wow flower arranging and singing whatever next??? Maybe a revolution like dancing and singing????	sarcasm
	Gah! Woke up and now having a headache	non-sarcasm

Table 7.3: Sample instances from evaluation datasets

- **LSTM with text**: In addition, we also implement a LSTM model for text classification. A popular way of modeling a sentence s for input into LSTM is to represent each word by a vector $x \in \mathbb{R}^d$ (Mikolov et al., 2013a), and sequentially input its word vectors $\langle x_1, x_2, ..., x_{|s|} \rangle$ to the LSTM model. As pre-trained word embeddings have shown to be useful in sarcasm detection with text (Ghosh et al., 2015; Zhang et al., 2016), we employ pre-trained GloVe word embeddings[2] (dimensions = 100, trained on the Wikipedia 2014 + Gigaword 5 corpus) (Pennington et al., 2014).

- **Riloff et al. (2013) contrast rules**: Riloff et al. (2013) popularized the notion of sarcasm as a contrast between positive and negative sentiment. We

[2]http://nlp.stanford.edu/projects/glove/

re-implement three of their rule-based algorithms for comparison: (i) *Positive sentiment only* (Pos only): an instance is labeled as sarcastic if it contains any positive term; (ii) *Negative sentiment only* (Neg only): an instance is labeled as sarcastic if it contains any negative term; and (iii) *Positive and Negative sentiment, unordered* (PosNeg unordered): an instance is labeled as sarcastic if it contains both a positive sentiment term and a negative sentiment term, in any order.

- **Positive-negative sentiment polarity (PosNeg)**: In the context of sarcasm detection, using sentiment polarities as features has been shown effective (Barbieri et al., 2014; Hernández-Farías et al., 2015; Joshi et al., 2015; Riloff et al., 2013). For this baseline method, we extract positive and negative sentiment scores from two existing lexicons: SentiWordNet (Baccianella et al., 2010) and NRC EmoLex (Mohammad and Turney, 2013). Then, we apply the same sequence models described earlier for emotions (§7.2.3), for comparison. The six merged LSTM encoders (§7.2.3.2) are replaced by two encoders for two parallel sequences of positive and negative sentiments.

All the approaches are evaluated using stratified 10-fold cross validation in order to maintain a similar ratio of class labels in the test set, especially useful in cases of imbalanced datasets such as these. The results are reported in terms of macro-averaged F-score (i.e., average over the two classes: *sarcasm* and *non-sarcasm*), where F-score is the harmonic mean of precision and recall.

Methods	$\mathbf{SASI_{am}}$		$\mathbf{SASI_{tw}}$	
	NB	SVM	NB	SVM
unigrams	0.569	**0.602**	0.571	0.551
uni + bigrams	0.584	0.513	0.563	0.576
uni + bi + trigrams	0.577	0.482	**0.584**	0.548
LSTM with text	0.592		0.554	

Table 7.4: Results of methods using text classification with lexical features only (without sentiment or emotion features).

7.3.3 Results

The macro-averaged F-scores of the proposed models and the various baselines applied to two evaluation datasets, SASI_{am} and SASI_{tw}, are analyzed below.

Table 7.4 compares the results of the methods that use text classification using only lexical information (i.e., without any affective information consisting of sentiment or emotion features). We observe that for the Amazon reviews, unigrams with SVM produces the better model, while for the Twitter data, unigrams + bigrams + trigrams with NB perform better. Additionally, when using n-grams data, simple traditional models such as SVM and NB perform better than the more complex neural network LSTM, most likely due to the fact that the LSTM needs much larger training datasets to perform effectively, than is usually available for sarcasm due to the rare positive class (Abercrombie and Hovy, 2016).

Table 7.5 summarizes the macro-averaged F-scores of the different models using features derived from sequence data, trained with NB and SVM. From the results,

Methods	$\mathbf{SASI}_{\text{am}}$		$\mathbf{SASI}_{\text{tw}}$	
	NB	SVM	NB	SVM
PosNeg BF	61.6	59.3	60.0	56.3
PosNeg BF + n-grams	62.2	47.4	63.9	59.8
PosNeg BF + EF	61.5	61.2	62.8	55.6
PosNeg BF + EF + n-grams	61.2	55.8	63.3	61.4
Average PosNeg	61.6	55.9	62.5	58.3
Emotions BF	62.5	63.8	**64.7**	64.6
Emotions BF + n-grams	63.4	58.8	60.2	63.3
Emotions BF + EF	64.7	57.5	62.2	52.5
Emotions BF + EF + n-grams	**65.8**	58.5	64.1	63.8
Average Emotions	64.1	59.7	62.8	61.1

Table 7.5: Results of classification using sequence-derived features (shown in % F-score). BF = Basic Features; EF = Extended Features. Best result for each dataset shown in **bold**.

we notice that the models using features derived from emotion sequences (from all three emotion resources: WNA, NRC and SECO) outperform those using positive-negative sentiment scores (obtained using two sentiment lexicons: SWN and NRC). On average, the Emotions model using six emotions outperforms the PosNeg model using two sentiment polarities, suggesting the usefulness of exploiting finer-grained affective information. For the Amazon reviews, the combination of Basic Features (BF), Extended Features (EF) and n-grams yields the best results, while for the

Methods	**SASI$_{\text{am}}$**	**SASI$_{\text{tw}}$**
Baselines		
n-grams	60.2 ($_{\text{unigrams}}$)	58.4 ($_{\text{uni+bi+trigrams}}$)
Riloff et al. (2013)		
(i) Pos only (s_1/ s_2)	51.9 / 32.1	57.2 / 33.8
(ii) Neg only (s_1/ s_2)	51.6 / 30.6	56.5 / 41.3
(iii) PosNeg unordered (s_1/ s_2)	50.2 / 35.2	54.6 / 41.7
Our proposed model ETS		
PosNeg (BF + n-grams)	62.2	63.9
PosNeg (HMM)	57.7	61.9
PosNeg (LSTM)	59.3	63.2
Emo (BF + EF + n-grams)	65.8	64.1
Emotions (HMM)	**66.6**	64.6
Emotions (LSTM)	65.3	**64.9**

Table 7.6: Results comparing our proposed models using sequences against baseline methods. The sentiment lexicons s_1 = NRC EmoLex; s_2 = SentiWordNet.

Twitter data, raw BF works well.

Lastly, Table 7.6 summarizes the overall results of our proposed models. Interestingly, contrary to the results of the inter-annotator agreement scores of the two datasets, the computational models perform better on SASI$_{\text{am}}$ than SASI$_{\text{tw}}$ dataset, likely due to the unconventional language that is generally used in the latter (tweets) yielding poorer emotion vectors.

Amongst the baseline methods, simple n-grams set a competitive baseline score,

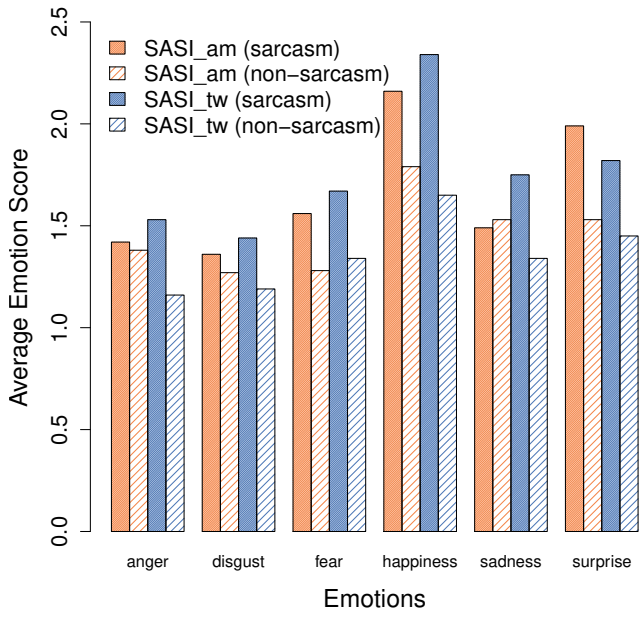
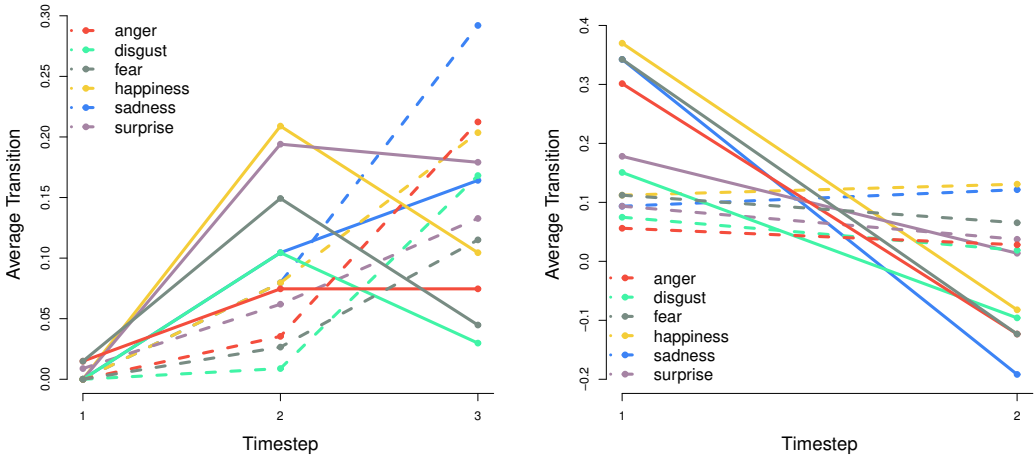


Figure 7.3: Distribution of emotions in two sarcasm datasets

consistent with previous results (Joshi et al., 2016b). Similarly, consistent with the results shown in previous research (Riloff et al., 2013), although somewhat counter-intuitive, the *Positive sentiment only* (Pos only) produces better results than using *Negative sentiment only* (Neg only) or using both *Positive and Negative sentiment, unordered* (PosNeg unordered).

Next, although all the PosNeg models employ the same two sentiment lexicons, our sequence models leveraging *transitions* of positive and negative scores improve over the models without the transitions. Finally, all the three models using emotion transitions outperform all the other models, indicating the effectiveness of leveraging *transitions of finer-grained emotions* in sarcasm detection.



(a) SASI_{am} (b) SASI_{tw}

Figure 7.4: Transitions of emotions between chunks. Solid lines = *sarcasm*; dotted lines = *non-sarcasm*. (a) 3 transitions between 4 chunks; (b) 2 transitions between 3 chunks.

7.4 Model Analysis

To further investigate the effects of various parameters influencing the performance of the proposed models, we conduct a series of experiments for model analysis.

7.4.1 Distribution and Transitions of Emotions

An overview of the approximate distribution of various emotions in the *sarcasm* and *non-sarcasm* classes of the two evaluation datasets is presented in Figure 7.3. Note that these emotion scores were obtained through unsupervised emotion labeling algorithm (described earlier in §7.2.2), and therefore, likely contain some degree of noise. However, even this noisy distribution lends one discriminative observation: on aver-

age, the sarcasm class exhibits more emotion than the non-sarcasm class for all the emotions for both the datasets, a distinction that is more noticeable in the SASI_{tw} dataset. Consistent with previous research (Sulis et al., 2016), we also find *happiness* to have the highest scores, while *disgust* is weakly represented across both the datasets.

Next, we analyze the transitions of these emotions over sequences of chunks (i.e., the difference in emotion scores between two consecutive chunks), as presented in Figure 7.4. It is observed that both the datasets exhibit distinctly different patterns of emotion transitions for the sarcasm versus the non-sarcasm class, which guides our hypothesis that the transitions of emotions could be a useful feature for detecting sarcasm. Different number of chunks work better on different types of data. Therefore, we present the transitions between 4 chunks for SASI_{am} and 3 chunks for SASI_{tw} dataset.

7.4.2 Effect of Number of Words per Chunk

One of the parameters in this study is k, the number of words per chunk in the set of chunks C constituting the sequence of each instance. Recall that as k increases, the length of the sequence, n, decreases. Here, $1 \leq k \leq \gamma$, where γ is the value of k when $n = 2$ (i.e., there are at least 2 elements in the sequence in order to establish emotion transitions). Table 7.7, which summarizes the best results obtained from three different ways of chunking (Emotions HMM model), indicates that the fixed-k approach is more suitable than phrase-based or equal-n chunking, and therefore, we adopt fixed-k chunking for all the experiments.

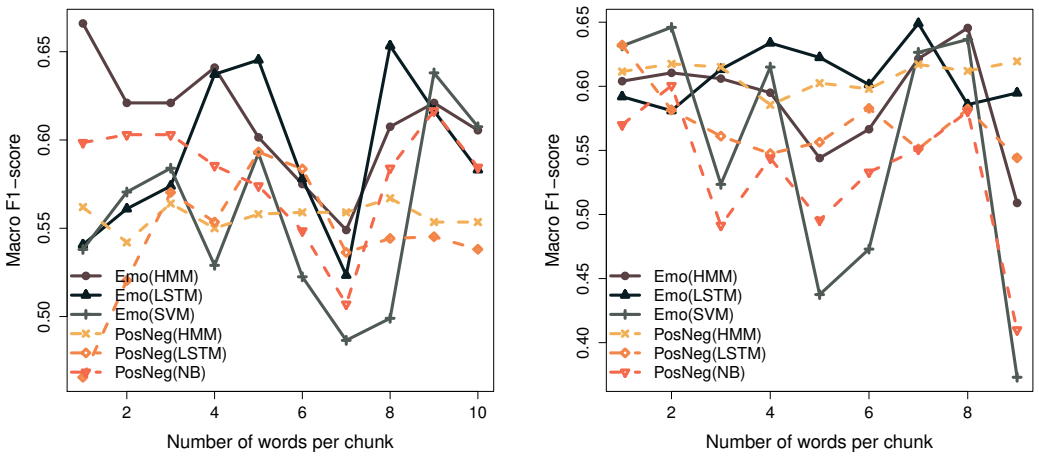
chunking	**SASI**$_{\text{am}}$	**SASI**$_{\text{tw}}$
best phrase-based	59.9	57.7
best equal-n	62.3	62.5
best fixed-k	66.6	64.6

Table 7.7: Results of different chunking

To further examine the effect of k in fixed-k chunking, we present the results obtained using different values of k in Figure 7.5. In particular, better results for SASI_{am} are obtained when $k = 1$, and for SASI_{tw} when $k = \gamma$ (i.e., $n = 2$), suggesting that for Amazon user reviews, the sequence models most effectively capture emotion transitions at word level (i.e., many chunks), whereas for tweets, sequences composed of fewer chunks (e.g., two chunks) are more suitable. This discrepancy likely reflects the inherent differences between the two different genres of text in the evaluation datasets. One shared observation, though, is that consecutive words grouped together into arbitrary chunks ($1 < k < \gamma$) are less useful, possibly because random concatenation of words into chunks is less likely to yield reasonable sequences.

7.4.3 Effect of Number of Emotions

To further study the effects of various emotions, we deconstruct Ekman's model of six emotions (Ekman, 1992) into its various subsets to examine if there exist particular emotions that may be more suitable for detecting sarcasm. Summarized in Table 7.8 are the results obtained using sequences of only one emotion at a time, with Emotions LSTM model. While *sadness* and *surprise* are the two most discriminative emotions



(a) SASI_{am} (b) SASI_{tw}

Figure 7.5: Results for different values of k in fixed-k chunking

for the Amazon reviews dataset, *anger* and *surprise* are the top two emotions for the tweets data. However, it seems that sequences of one emotion are not sufficient enough as the superset of all six emotions is still the best model in this configuration.

Further, Table 7.9 presents the results of the top four best subsets (out of the fifteen possible combinations) consisting of two emotions each. Considerable improvement is noticed when using sequences of two emotions for both the datasets compared to sequences of single emotions. In fact, for SASI_{tw} dataset, the simple combination of two emotions (*anger* and *surprise*) yields an improvement over the results obtained from using the superset of all six emotions.

Similarly, out of the twenty possible subsets comprised of three emotions each, the results of the top four best subsets are presented in Table 7.10. While subsets of three emotions seem to be better than subsets of two emotions for SASI_{tw} on average,

emotions	F-score
Ekman's 6 emotions	**65.33**
sadness	60.55
surprise	60.35
anger	59.18
fear	57.22
happiness	57.07
disgust	55.51

emotions	F-score
Ekman's 6 emotions	**64.93**
anger	59.16
surprise	58.93
happiness	58.45
fear	58.28
sadness	56.24
disgust	51.69

Table 7.8: Results of individual emotions. (a) SASI_{am} (b) SASI_{tw}

emotions	F-score
Ekman's 6 emotions	**65.33**
sadness + surprise	62.31
anger + sadness	62.24
anger + surprise	61.89
fear + surprise	61.87

emotions	F-score
Ekman's 6 emotions	64.93
anger + surprise	**65.14**
happiness + surprise	62.92
digust + happiness	62.57
anger + happiness	61.71

Table 7.9: Results of top four best subsets of two emotions. (a) SASI_{am} (b) SASI_{tw}

the opposite is true for SASI_{am}.

The top results of combinations of four and five emotions, summarized in Tables 7.11 and 7.12, respectively, suggest that while SASI_{tw} benefits with a larger subset of emotions, SASI_{am} does best with subsets of two emotions at most or all six emotions. Interestingly, the results of SASI_{tw} in Table 7.11 show that the overall best results are obtained using a subset of four relatively negative emotions (*anger*, *disgust*, *fear* and *sadness*). Lastly, based on the results of Table 7.8 as well as Table 7.12, it seems that *disgust* is the least discriminative emotion for both domains of text.

emotions	F-score
Ekman's 6 emotions	**65.33**
fear + happiness + surprise	63.19
anger + happiness + surprise	60.73
anger + disgust + sadness	60.67
disgust + happiness + surprise	60.63

emotions	F-score
Ekman's 6 emotions	**64.93**
anger + disgust + happiness	64.86
anger + happiness + sadness	63.91
anger + fear + sadness	63.30
anger + happiness + surprise	62.74

Table 7.10: Results of top four best subsets of three emotions. (a) SASI_{am} (b) SASI_{tw}

emotions	F-score
Ekman's 6 emotions	**65.33**
fear + happiness + sadness + surprise	61.87
anger + fear + happiness + surprise	61.19
anger + disgust + fear + surprise	60.68
anger + fear + sadness + surprise	60.39

emotions	F-score
Ekman's 6 emotions	64.93
anger + disgust + fear + sadness	**65.62**
anger + fear + happiness + sadness	63.85
anger + happiness + sadness + surprise	63.72
anger + fear + sadness + surprise	62.96

Table 7.11: Results of top four best subsets of four emotions. (a) SASI_{am} (b) SASI_{tw}

7.5 Comparing ETS and AWES

In chapter 6, we introduced a model, called AWES, for computational sarcasm detection by harnessing features from affective word representations. The sarcasm detection model described in this chapter exploits transitions between emotion sequences. In essence, the former model employs document-level features whereas the latter comprises of features designed at intra-document-level, i.e., chunks of document.

Now, we compare the performance of these two design methodologies under similar

	emotions	F-score
$SASI_{am}$	Ekman's 6 emotions	65.33
	Ekman's 6 emotions - *disgust*	60.75
$SASI_{tw}$	Ekman's 6 emotions -	64.93
	Ekman's 6 emotions - *disgust*	64.49

Table 7.12: Results of top subsets of five emotions (Ekman's all six but one)

Methods	**$SASI_{am}$**	**$SASI_{tw}$**
AWES-senti	0.61	0.57
AWES-emo	0.64	0.55
ETS	**0.67**	**0.65**

Table 7.13: Comparing the performances of ETS and AWES.

settings. The results summarized in Table 7.13, in terms of macro F-score, indicate that the transitions of emotions within documents offer additional improvement over using discrete document-level features such as the AWES model.

7.6 Conclusions

In this chapter, we introduced a novel methodology for detecting sarcasm by formulating it as an emotion sequence classification problem by leveraging the shifts in emotion patterns. To demonstrate the potential of our approach, we conducted experiments on two evaluation datasets, where the proposed approach (emotion sequences with HMM and LSTM) outperformed several baselines.

In particular, the results indicate that exploiting finer-grained categories of emotions such as *happiness*, *sadness*, etc., yield better results than binary polarities comprising of positive and negative sentiment. Furthermore, sequences of emotion transi-

tions offer additional improvement in performance. Of the six emotions from Ekman (1992) model that were studied in this research, *sadness* was found to be the most discriminative emotion for SASI_{am}, and *anger* for SASI_{tw}, while for both the datasets, *disgust* was found to be the least discriminative. Furthermore, while the tweets data benefited with a larger subset of emotions, on average the Amazon reviews were better off with subsets of two emotions or all six emotions. Lastly, a comparative experiment demonstrated that intra-document features such as transitions within emotion sequences is more beneficial for sarcasm detection than entire document-level features such as previously used in the AWES model.

Chapter 8

Predicting Helpfulness of Online Reviews

Leading up to this chapter, we have introduced and analyzed numerous models for augmenting affect analysis in text, particularly through the lens of emotion detection and sarcasm detection. While functional on their own as stand-alone emotion analysis and sarcasm recognition systems, these affective models can further enhance numerous application systems. In this chapter, we first discuss various applications benefiting from integration of affective models. Then, we present a case-study describing the application of one of our affective models in a non-affective framework in order to demonstrate the usefulness of the proposed model in tasks beyond affect detection. Specifically, we leverage emotion-enriched word representations (described earlier in chapter 5) to predict the helpfulness of online reviews.

The chapter is structured as follows. In Section 8.1, a general discussion on various applications leveraging affective information is presented. Then, Section 8.2

introduces our case-study involving one specific application domain: predicting helpfulness of reviews.

8.1 Applications of Affective Systems

Numerous applications benefit from affective information (Mohammad and Turney, 2013). While a brief overview of several applications was presented earlier in chapter 1, in this chapter we describe specific instances of some of the applications.

- Financial decision-making: De Bondt et al. (2013) investigated the specific role of emotion information in predicting everyday financial decisions under the framework of behavioral research. Along similar lines, Ahmed (2017) explored the correlation between a financial event in the form of stock price movement and the degree of emotion.

- Suicide prevention: A major public health concern worldwide, successful suicide prevention requires adequate suicide risk assessment. Although online platforms are increasingly used for expressing suicidal thoughts, manual monitoring is infeasible given the huge amount of data. Therefore, Desmet and Hoste (2013) used an automatic emotion detection system to pinpoint 15 different emotions which may be indicative of suicidal behavior, with the best performance achieved for six most frequent emotions including thankfulness, guilt, love, information, hopelessness and instructions. The study concluded that natural language processing techniques involving emotion analysis have considerable application potential for suicide prevention.

- Customer relations: Prompt and knowledgeable responses to customer emails, which often contain complaints about negligence, incompetence, unfriendliness and unresponsive personnel, are critical in maximizing customer satisfaction. Referring to such customer emails as emotional emails, Gupta et al. (2013) described a method for extracting salient features and identifying emotional emails in customer care, which considerably improved the system performance.

- Intelligent tutoring systems: Emotion, or affect, plays a central role in learning. In particular, promoting positive emotions such as engagement and enjoyment is important for student motivation. Wiggins et al. (2014) analyzed a dataset of student facial videos from computer-mediated human tutorial sessions for understanding nonverbal behaviors. To improve the emotional interaction between learners and the tutoring system, Sun et al. (2013) introduced an emotional interaction agent which can deduce users' emotional statuses, provide help when needed and mark emotional difficulty of the learned pedagogical units.

- Text-to-speech (TTS): Trilla and Alias (2013) adapted a sentiment analysis system for expressive speech synthesis in order to improve the state-of-the-art in text-to-speech studies.

- Human-computer interaction (HCI): Speech data contains a high information density, where speech characteristics can reveal the speaker's emotion, intention and motivation. While HCI engineers rely on technically measurable acoustic and spectral features, psychologists analyzing emotions work in emotion categories, schemes or dimensional emotional spaces (Hartmann et al., 2013). With an aim of combining the two essential perceptions, Hartmann et al. (2013) intro-

duced a system for emotion detection from speech to allow machines to allocate affective states during HCI. In similar vein, Siegert et al. (2013) presented a mood model that incorporates personality traits based on emotionally labeled data in order to recognize changes in the users emotional reaction during interaction under the framework of HCI.

- Music: Huang and Lin (2013) integrated an emotion-based algorithmic composition mechanism into generative music. Their technique generates emotion music according to the scenario as well as plays different content of the music every time to make listeners feel "fresh". The results indicated that the proposed method was successful at generating music emotions such as happy, angry, sad and joy. Furthermore, they postulated that their proposed idea could be extended to fields including multimedia and game to automatically generate the background music according to the interaction between human and machine.

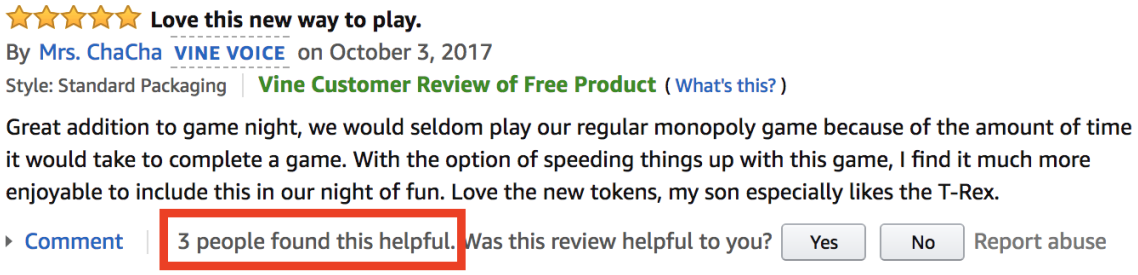
- Sarcasm from student feedback: Altrabsheh et al. (2015) addressed the issue of sarcasm detection in an educational context. Specifically, they applied sarcasm recognition models to identify sarcasm from student feedback collected via Twitter.

- Disease progression: Larsen et al. (2016) investigated the role of complex social cognition including emotion recognition and sarcasm detection in Huntington disease (HD). Their findings support a theory of impaired social-congitive functions in the early stages of HD. Test performances decreased with increasing disease burden, suggesting that social-congitive tests such as simple emotion recognition may be useful for tracking disease progression.

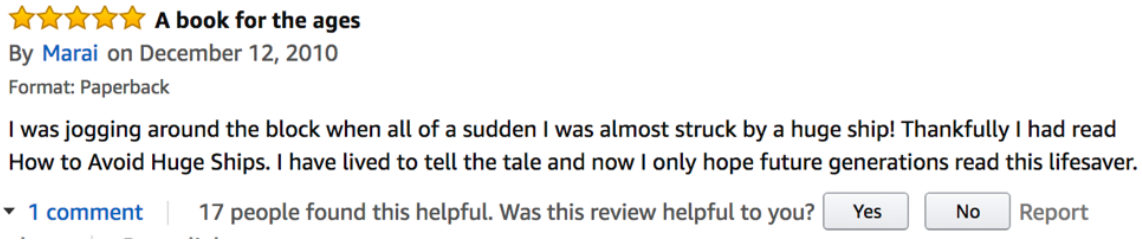
8.2 Predicting Helpfulness of Reviews

These days, people's decision-making process is strongly influenced by online reviews. The helpfulness of a review is typically shaped by the wisdom of the crowd. Predicting the helpfulness of reviews, therefore, can help to save considerable time and find popularly useful suggestions quickly and easily. Most previous works focus on exploring new features with external data source, such as user profile, semantic dictionaries, and so on, which may require substantial feature engineering. Most recently, one study showed the effectiveness of using information derived from word representations which eliminates the need extensive feature engineering. However, reviews are also known to contain significant expressions of affect such as sentiments, emotions and even sarcasm. Figure 8.1 shows two examples of Amazon reviews, with (a) conveying positive emotion and (b) expressing sarcasm.

In this chapter, we investigate whether models of affective representations presented earlier in this thesis can assist in non-affective tasks such as predicting the helpfulness of reviews. Helpfulness can be computed in various ways. Amazon reviews data is generally accompanied with information such as the number of people who voted a given review as helpful (h) along with the total number of people who voted (t). In such a case, we establish the helpfulness score as the ratio of helpful votes to total votes, i.e., h/t. The higher the score, the more helpful the review is ascertained to be.



(a) expression of positive emotion



(b) expression of sarcasm

Figure 8.1: Examples of Amazon reviews

8.2.1 Related Work

The task of estimating and predicting text helpfulness has been addressed in several works. Early work in predicting the helpfulness of online reviews considered the number of superlative and comparative adjectives and adverbs appearing in a review and applied a regression model to predict the review helpfulness (Zhang and Varadarajan, 2006), where the experimental results indicated shallow syntactic features to be one of the most influential predictors. Kim et al. (2006) also applied regression to predict the helpfulness of product reviews based on various classes of lexical, structural, syntactic, semantic and meta-data related features such as review length, unigrams and

product ratings. Text surface features and unigrams proved to be the most helpful features and have been widely used in later researches. Liu et al. (2008) described a model to compute review quality in the movie domain, where three distinctive features were found to influence the review quality, namely: (a) review expertise, (b) writing style, and (c) review timeliness.

The approaches described above focus only on textual features of reviews; other approaches suggest to include social information to predict review quality (Lu et al., 2010; O'Mahony and Smyth, 2009). While O'Mahony and Smyth (2009) calculated review quality in TripAdvisor by building a user-hotel bipartite graph and investigating its structural features, Lu et al. (2010) suggested to explore the social network of a reviewer to get some insights about his/her expertise guided by the assumption that a user $u1$ will admit a user $u2$ in his/her social network if and only if the quality of reviews posted by $u2$ is at least as high as those posted by $u1$. Meo et al. (2017) relied on explicit trust links to compute user reputation, where the reputation of a user is understood as his/her ability of posting helpful reviews.

Most of the previously described works focus on designing efficient features, in particular external features such as date (Liu et al., 2008), product rating (Kim et al., 2006) and product type (Mudambi and Schuff, 2010). Compared to external features, intrinsic features such as semantic dictionaries (Yang et al., 2015) and emotional dictionaries (Martin and Pu, 2014), can provide further insights and explanations for the prediction results, and support better cross-domain generalization. Liu et al. (2017) investigated a new form of intrinsic features: the argument features. An argument is a basic unit people use to persuade their audiences to accept a particular state of affairs. It usually consists of a claim (also known as conclusion) and some

premises (also known as evidences) offered in support of the claim.

The inclusion of other intrinsic features such as the readability, subjectivity and emotion (Martin and Pu, 2014; O'Mahony and Smyth, 2010) further improved models of helpfulness. Yang et al. (2015) hypothesized that helpfulness is an internal property of text and introduced semantic features to model the review text. Their experiments demonstrated that a model built purely from text is transferable between different product categories. Yang et al. (2016) solved the task at a deeper level by understanding the content of reviews by integrating aspect information using topic modeling into helpfulness prediction.

Another strand of research attempted to solve this problem from a decision-making perspective (Danescu-Niculescu-Mizil et al., 2009; Mudambi and Schuff, 2010), where a helpful review was defined as a peer-generated product evaluation that could facilitate the consumers' purchase decision process. In particular, Mudambi and Schuff (2010) revealed that review rating, product type and word counts are three important factors for a helpful review and modeled the helpfulness score as a combination of these three factors. Danescu-Niculescu-Mizil et al. (2009) studied the option evaluation from Amazon book reviews and found that the final helpfulness voting for a review is affected by many external factors, including the ratio of its helpfulness score to those of other reviews. Liu et al. (2007) worked on how to detect low quality reviews. They introduced features to model the informativeness, subjectiveness and readability of a review and classified them into high or low qualities.

An unsupervised algorithm, RevRank, was proposed to rank the helpfulness of online book reviews (Tsur and Rappoport, 2009). First, a lexicon of dominant terms across reviews was constructed, and then, a virtual core review based on this lexicon

was created. Finally, the distance between the virtual review and each real review was used to determine the overall helpfulness ranking. Hong et al. (2012) developed a binary helpfulness classification system using a set of novel features based on needs fulfillment, information reliability and sentiment divergence measure, with their system outperforming some earlier researches with the same dataset. Lee and Choeh (2014) proposed a helpfulness prediction neural network model and made use of products, review characteristics, and reviewer information as features. This was the first study to predict helpfulness using neural networks, with the proposed model outperforming the conventional linear regression model analysis in predicting helpfulness. Zhang et al. (2015) proposed a comment-based collaborative filtering approach which captures correlations between hidden aspects in review comments and numeric ratings. They also estimated the aspects of comments based on profiles of users and items, evaluated on a Chinese review dataset. Krishnamoorthy (2015) proposed a predictive model that extracts novel linguistic category features by analyzing the textual content of review. The author made use of review metadata, subjectivity and readability related features for helpfulness prediction and showed that the proposed linguistic category features were better predictors of review helpfulness.

Most similar to our application is the work of Chen et al. (2016), which introduced word embedding features to model review documents. However, they only considered contextual word embedding models without any affective knowledge. Instead, we apply our emotion-enriched word representations in order to augment models of review helpfulness prediction by exploiting affective knowledge which is an integral component of product reviews.

8.2.2 Proposed Model

Assume $\mathcal{D} = \{w_1, w_2, ..., w_n\}$ denotes a review document, and let $V\ (\mathcal{D}\ ) = \{h, t\}$ denote the helpfulness rating of the review, where h denotes the number of people who voted review $\mathcal{D}$ to be helpful and t denotes the total number of votes. Given $\mathcal{D}$, the task is to predict whether the review is helpful or not.

Helpfulness can be defined in a number of ways. Assuming the helpfulness score to be a numeric value between 0 and 1, with 0 being the least helpful a score can be and 1 being the most helpful, $\mathcal{H} = [\,0, 1\,]$.

The following steps describe the process of applying our proposed emotion-enriched word representations (introduced earlier in chapter 5) as features for augmenting models of review helpfulness prediction:

1. Let $E \in \mathbb{R}^{d \times |\mathcal{V}|}$ denote the embedding matrix of emotion-enriched word representations, where $\mathcal{V} = \left\{w_1, ..., w_{|\mathcal{V}|}\right\}$ is the vocabulary and each word w_i is represented as a d-dimensional continuous vector $\mathbf{x}_i \in \mathbb{R}^d$. For each word w_i in $\mathcal{D}$, we obtain its vector representation $\mathbf{x}_i = \langle x_{i1}, x_{i2}, ..., x_{id} \rangle$ by looking up in the matrix E. Note that the full procedure of obtaining E was described earlier in chapter 5.

2. For each $\mathcal{D}$, we obtain a fixed-length document-level representation $\mathbf{D}$ by computing the average of all the word vectors along each dimension. For $j \in \{1, ..., d\}$, the jth dimension of $\mathbf{D}$ is:

$$\mathbf{D}_j = \frac{\sum_{i=1}^{n} x_{ij}}{n} \tag{8.1}$$

where n is the number of words in $\mathcal{D}$, and $\mathbf{x}_i \in \mathbb{R}^d$ denotes the word embedding vector of the ith word.

3. Finally, $\mathbf{D}$ is fed into a supervised classification algorithm to predict the helpfulness of $\mathcal{D}$.

8.2.3 Experiments

In this section, we first present the evaluation dataset and the baselines used for comparison. Then, we discuss the results of the experiments.

8.2.4 Evaluation Dataset

In order to predict review helpfulness, we draw on the Amazon review dataset (McAuley et al., 2015), consisting of product reviews for a variety of product categories spanning almost eighteen years from May 1996 to July 2014.

For this Amazon product reviews data, we compute $\mathcal{H}$ as follows: $\mathcal{H} = \frac{h}{t}$, where $h \leq t$, $t \geq k$, and k denotes some threshold for the minimum number of total votes a review should have received in order to qualify for consideration. This threshold is necessary to limit the number of biased or noisy reviews with just one or two votes.

An example of a review (in json format) is presented in Figure 8.2. The noteworthy fields of interest include "*helpful*" and "*reviewText*". In this example, 6 out of 7 people found the review to be helpful, i.e., $h = 6$, $t = 7$, and therefore, helpfulness score $\mathcal{H} = 0.857$. Additionally, k is set to 5.

As there can be a considerable amount of noise in natural language text data such as user reviews, we pre-process the reviews and filter out those reviews containing

{"reviewerID": "A00001483M88NBD66LEP0", "asin": "B004WPCQKG", "review
erName": "JARPD", "helpful": [6, 7], "reviewText": "No matter what we
did the bills just kept jamming in the machine. The bill counts were
not consistent. We returned the product because it did not work corr
ectly.", "overall": 1.0, "summary": "Disappointed", "unixReviewTime":
1353283200, "reviewTime": "11 19, 2012"}

Figure 8.2: An example of an Amazon review in json format

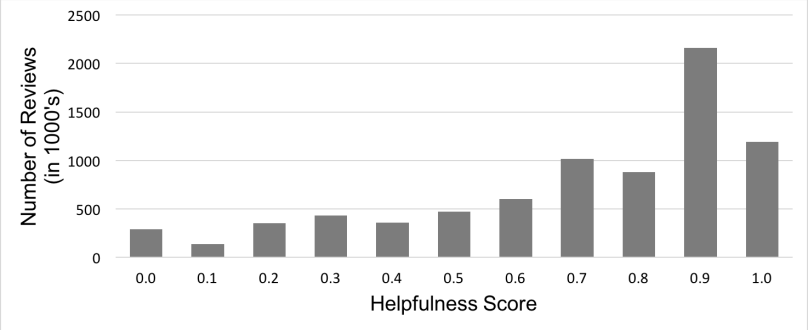


Figure 8.3: Distribution of helpfulness scores

less than 5 words. Out of the total 80 million reviews in the dataset, this leaves us with 10.5 million reviews. The distribution of reviews with respect to helpfulness score is shown in Figure 8.3. In the graph, the data appears to be skewed to the left left suggesting that more number of reviews are generally found to be helpful than otherwise.

We cast the review prediction problem into a binary classification problem, and randomly select 1000 helpful reviews ($\mathcal{H} = 1$) and 1000 non-helpful reviews ($\mathcal{H} = 0$), comprising the two classes, for constructing our evaluation dataset for the experiments.

8.2.4.1 Baselines

The proposed approach of applying emotion-enriched word representations (EWE) as features is compared against several baselines including the following features:

- **Unigrams**: Unigram features have proven effective in review helpfulness prediction (Chen et al., 2016; Yang et al., 2015). After removing all the stopwords and words occurring less than 10 times in the total corpus, a word dictionary $\{w_1, w_2, ..., w_n\}$ is built, and each review r_i is represented by a vector $\mathbf{r}_i = \{v_1, v_2, ..., v_k\}$, where v_i is the term frequency-inverse document frequency (*tf-idf*) weight of word w_j in review r_i.

  One of most the popular term weighting schemes, *tf-idf*, is a numerical statistic that is intended to reflect how important a word t is to a document in a collection or corpus (Rajaraman and Ullman, 2012). Typically, the *tf-idf* weight is composed of two terms: term frequency (tf) and inverse document frequency (idf).

  Term frequency is the number of times a word t appears in a document d divided by the total number of words in that document (as a way to normalize with respect to document length). Formally, tf(t) is calculated as follows:

$$tf(t) = \frac{\text{number of times } t \text{ appears in } d}{\text{total number of terms in } d} \tag{8.2}$$

  Inverse document frequency, which measures how important a term is, is computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the term appears. Specifically, idf(t) is

computed as:

$$idf(t) = \log \frac{\text{total number of documents}}{\text{number of documents with } t \text{ in it}} \tag{8.3}$$

Finally, the *tf-idf* weight of t is:

$$tf\text{-}idf(t) = tf(t) \cdot idf(t) \tag{8.4}$$

- **Surface features**: Following previous research (Chen et al., 2016; Yang et al., 2015), this baseline consists of textual surface features such as the number of sentences in the review, the number of words in the review, the average length of sentences, the number of exclamation marks, the percentage of question sentences and the ratio of uppercase to lowercase characters in the review text.

- **Generic word embeddings**: Contextual word representations such as word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) are used to derive document-level features. Each document is represented in terms of the average of the word vectors of all its words.

8.2.4.2 Results

The performance of the various features in predicting review helpfulness, summarized in Table 8.1, is reported as macro F-score and evaluated in terms of 10-fold cross validation using Logistic Regression for supervised classification. The results indicate that simple unigrams baseline set a competitive baseline and surprisingly, the more extensive feature set of surface features does not bring any additional improvement. However, word representations features are considerably better than simple textual

Methods	Macro F-score
Unigrams	0.699
Surface features	0.629
word2vec	0.711
GloVe	0.727
EWE	**0.734**

Table 8.1: Results of various features in review helpfulness prediction.

features, with the proposed emotion-enriched word representations (EWE) yielding the overall best results.

8.3 Conclusions

This chapter illustrates the usefulness of affective information in the form of emotion-enriched word representations in a non-affective task such as review helpfulness prediction. The proposed application of emotion-enriched representations outperforms several baselines demonstrating that affective word representations can be effectively applied to solve problems beyond affect detection.

Chapter 9

Conclusions and Future Directions

Affect detection from text is the problem of identifying affective states such as sentiment, emotions, sarcasm and so on from natural language data such as reviews, tweets and many more. Throughout this thesis, we augmented affect detection centered around two particular modules - emotion detection and sarcasm detection. In a nutshell, this work attempts to identify computational patterns for automatic classification of emotion and sarcasm from natural language text which is generally wrinkled with oddities.

This chapter concludes the work described in this dissertation, with a brief summary of contributions presented in section 9.1 and avenues of future work discussed in section 9.2.

9.1 Summary of Contributions

In this dissertation, we proposed and analyzed several algorithms for improving emotion detection and sarcasm detection from text documents. In conclusion, the key

contributions of our work can be summarized as follows:

- We presented a novel unsupervised approach for automatically learning word-emotion association scores (chapter 4), which are the basic foundation of many emotion detection algorithms, especially the unsupervised ones. We demonstrated that our proposed approach of selective co-occurrences involving positional context (especially *preceding*) and mutual exclusivity (where a word is primarily strongly associated with only one emotion category in a given window of context) is beneficial for obtaining relevant scores, thereby improving the overall accuracy of emotion classification.

- We described a novel method of learning emotion-enriched word representations[1] (chapter 5), where words with similar emotion connotation are projected into neighboring regions of an n-dimensional affective embedding space. Our emotion-rich embeddings showed considerable improvements over traditional generic embeddings in tasks of emotion classification and emotion similarity.

- We introduced a novel model for detecting sarcasm in text by leveraging affective word representations[2] obtained from weakly labeled data (chapter 6). Extensive evaluation on six sarcasm datasets of short and long text documents demonstrated the effectiveness of the proposed framework. In particular, it was observed that sentiment word representations are more suitable for short text documents such as tweets, whereas emotion word representations benefit sarcasm detection in long documents such as product reviews and discussion

[1] Available for research upon request.

[2] Available for research upon request.

posts.

- We introduced another model for detecting sarcasm in text by formulating it as an emotion sequence classification problem by leveraging the shifts in emotion patterns (chapter 7). Of the six emotions from Ekman (1992) model that were studied in this research, *sadness* and *anger* were found to be the two most discriminative emotions, while *disgust* was found to be the least discriminative emotion for sarcasm detection.

- Lastly, we demonstrated the usefulness of one of our affective models in a non-affective setting by designing a model for predicting the helpfulness of online reviews (chapter 8).

9.2 Future Directions

The work presented in this dissertation outlines two key axes (emotion detection and sarcasm detection) along which affect detection in text is augmented. While the affective algorithms presented in this thesis represent significant advances in detecting emotion and sarcasm from text documents, there are still many interesting opportunities that deserve further pursuit.

- **Diverse Set of Seed Words**:

  In chapter 4, we introduced a novel approach to obtaining word-emotion association scores from large unlabeled text corpora. While initializing the process with as little as one seed word per emotion category enables the process to remain as unsupervised as possible, exploring a more diverse set of seed words

could bring further improvements to the word-emotion association algorithm, especially for the emotion categories such as *surprise* which are particularly difficult to classify correctly.

- **Alternate Models of Emotions**:

  In chapter 5, we described a novel method of learning emotion-aware word representations, grounded in the widely adopted Ekman (1992) model of six emotions. Similarly, in chapter 7, we described a model for detecting sarcasm using affective knowledge representations modeled along the spectrum of Ekman's model of emotions. Wide-ranging research in psychology has proposed many different taxonomies of emotions such as categorical (Plutchik (1980) model of eight emotions), or dimensional models consisting of valence, arousal and dominance. Some interesting lines of future work could include considering these alternate models of emotions for learning affective representations.

- **From Words to Phrases**:

  Chapters 5 and 6 proposed neural network architectures for learning affective representations at word level. However, certain affective expressions typically described at phrase level cannot be captured by word-level analysis. For instance, consider the following examples:

  1. *Eva will go off the deep end if her kids leave the kitchen in a mess again.*[3]
  2. *I can't believe this went on for so long, and we were blissfully unaware of it.*

[3]The idiom "*go off the deep end*" means to become angry or upset.

3. *The news brought them little happiness.*

Emotion is often conveyed by longer units of text or by phrases, for example, the idiom "go off the deep end" in (1), or the expression "blissfully unaware" in (2), or the linguistic feature serving to modify emotion "little happiness" in (3) (Aman and Szpakowicz, 2007). One possible future work could explore the learning of affective representations at phrase level.

- **Enhanced Embeddings**:

  One possible way of further enhancing the text representations, or specifically word embeddings, proposed in this thesis includes considering network or graph structures, where network-based lexical databases (e.g., WordNet) could be leveraged for acquiring contexts of words. Another possibility includes exploring dynamic embeddings (Yao et al., 2018), where the learned embeddings can be tweaked or updated with respect to evolving connotation of words over time. Yet another avenue of future work could explore topic-enhanced word embeddings (Ren et al., 2016), where topic information is incorporated along with affective information in order to generate more coherent text representations.

- **Multilingual Data**:

  While the proposed models in this thesis were evaluated only on English datasets, most of the models are language-agnostic. For example, the emotion-enriched and sentiment-specific embeddings from chapters 5 and 6 could be generated from source corpus originating in another language. With recent easy availability of numerous multilingual datasets (Giatsoglou et al., 2017; Kumar et al.,

2017; Mikula et al., 2017; Nguyen et al., 2017; Tocoglu and Alpkocak, 2018), it would be interesting to experiment with data from different languages in future work.

- **Notions of Co-occurrences**:

  A critical component underlying most models proposed in this thesis includes the basic notion of a window of text. There are many ways of defining a window such as a fixed number of words within a sentence, or all the words in a sentence, or extend the concept of a window to an entire document, or going beyond simple surface-level contexts to take into account syntax dependency parse trees, to name just a few. Future work exploring different notions of co-occurrences or windows of text could lend useful insights.

- **Unifying Spectrums of Affect**:

  The work in thesis is a first step towards illustrating the connections and inter-dependencies between two spectrums of affect, namely emotion and sarcasm. Ameur et al. (2017) most recently explored a component of this idea by utilizing emotional vector representations in the task of sentiment lexicon enrichment. Unifying the various spectrums of affect may provide significant additional benefits, which we leave for future exploration.

Bibliography

Abercrombie, G. and Hovy, D. (2016). Putting sarcasm detection into context: The effects of class imbalance and manual labelling on supervised machine classification of twitter conversations. *ACL 2016*, page 107.

Agrawal, A. and An, A. (2012). Unsupervised emotion detection from text using semantic and syntactic relations. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '12, pages 346–353, Washington, DC, USA. IEEE Computer Society.

Agrawal, A. and An, A. (2013). Kea: Expression-level sentiment analysis from twitter data. In *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2013, Atlanta, Georgia, USA, June 14-15, 2013*, pages 530–534.

Agrawal, A. and An, A. (2014). Kea: Sentiment analysis of phrases within short texts. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014.*, pages 380–384.

Agrawal, A. and An, A. (2016). Selective cooccurrences for emotion detection. In *Proceedings of COLING 2016*, Osaka, Japan.

Agrawal, A., Sahdev, R., Davoudi, H., Khonsari, F., An, A., and McGrath, S. (2016). Detecting the magnitude of events from news articles. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2016, Omaha, NE, USA, October 13-16, 2016*, pages 177–184.

Ahmed, S. M. (2017). Quantification of investor emotion in financial news by analyzing the stock price reaction. In *Information and Communication Technologies (ICICT), 2017 International Conference on*, pages 119–123. IEEE.

Alm, C. O., Roth, D., and Sproat, R. (2005). Emotions from text: Machine learning for text-based emotion prediction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 579–586, Stroudsburg, PA, USA. ACL.

Alm, E. C. O. (2008). *Affect in Text and Speech*. PhD thesis, University of Illinois at Urbana-Champaign.

Altarriba, J., Bauer, L. M., and Benvenuto, C. (1999). Concreteness, context availability, and imageability ratings and word associations for abstract, concrete, and emotion words. *Behavior Research Methods, Instruments, & Computers*, 31(4):578–602.

Altrabsheh, N., Cocea, M., and Fallahkhair, S. (2015). *Detecting Sarcasm from Students' Feedback in Twitter*, pages 551–555.

Aman, S. and Szpakowicz, S. (2007). Identifying expressions of emotion in text. In *Proceedings of the 10th International Conference on Text, Speech and Dialogue*.

Aman, S. and Szpakowicz, S. (2008). Using rogets thesaurus for fine-grained emotion recognition. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP*, pages 296–302.

Ameur, H., Jamoussi, S., and Hamadou, A. B. (2017). Sentiment lexicon enrichment using emotional vector representation. In *Computer Systems and Applications (AICCSA), 2017 IEEE/ACS 14th International Conference on*, pages 951–958. IEEE.

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0. In *LREC'10*, Valletta, Malta. ELRA.

Bamman, D. and Smith, N. A. (2015). Contextualized sarcasm detection on twitter. In *AAAI*.

Barbieri, F., Saggion, H., and Ronzano, F. (2014). Modelling sarcasm in twitter, a novel approach. *ACL 2014*, page 50.

Beeferman, D., Berger, A., and Lafferty, J. (1997). A model of lexical attraction and repulsion. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98, pages 373–380, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Bian, J., Gao, B., and Liu, T.-Y. (2014). *Knowledge-Powered Deep Learning for Word Embedding*, pages 132–148. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40.

Boylan, J. and Katz, A. N. (2013). Ironic expression can simultaneously enhance and dilute perception of criticism. *Discourse Processes*, 50(3):187–209.

Brosseau-Villeneuve, B., Nie, J.-Y., and Kando, N. (2010). Towards an optimal weighting of context words based on distance. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 107–115. Association for Computational Linguistics.

Budiu, R., Royer, C., and Pirolli, P. (2007). Modeling information scent: A comparison of lsa, pmi and glsa similarity measures on common tests and corpora. In *Large scale semantic access to content (text, image, video, and sound)*, pages 314–332. Le Centre De Hautes Etudes Internationales D'Informatique Documentaire.

Campbell, J. D. and Katz, A. N. (2012). Are there necessary conditions for inducing a sense of sarcastic irony? *Discourse Processes*.

Chaffar, S. and Inkpen, D. (2011). Using a heterogeneous dataset for emotion analysis in text. In *Proceedings of the 24th Canadian Conference on Advances in Artificial Intelligence*, Canadian AI'11, pages 62–67, Berlin, Heidelberg. Springer-Verlag.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27.

Chaumartin, F.-R. (2007). Upar7: A knowledge-based system for headline sentiment tagging. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 422–425, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chen, J., Zhang, C., and Niu, Z. (2016). *Identifying Helpful Online Reviews with Word Embedding Features*, pages 123–133. Springer International Publishing, Cham.

Chollet, F. et al. (2015). Keras. https://github.com/fchollet/keras.

Choudhury, M. D., Counts, S., and Gamon, M. (2012). Not all moods are created equal! exploring human emotional states in social media. In *Proceedings of the Sixth International Conference on Weblogs and Social Media, Dublin, Ireland, June 4-7, 2012*.

Chowdhuri, A. D. and Bojewar, S. (2016). Emotion detection analysis through tone of user: A survey. *Emotion*, 5(5).

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*.

Colston, H. L. (1997). Salting a wound or sugaring a pill: The pragmatic functions of ironic criticism. *Discourse Processes*, 23(1):25–45.

Danescu-Niculescu-Mizil, C., Kossinets, G., Kleinberg, J., and Lee, L. (2009). How opinions are received by online communities: A case study on amazon.com helpfulness votes. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 141–150, New York, NY, USA. ACM.

Danisman, T. and Alpkocak, A. (2008). Feeler: Emotion classification of text using vector space model. In *AISB 2008 Convention, Communication, Interaction and Social Intelligence*, volume vol. 2, Aberdeen, UK.

Davidov, D., Tsur, O., and Rappoport, A. (2010). Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *CONLL '10*. ACL.

De Bondt, W., Mayoral, R. M., and Vallelado, E. (2013). Behavioral decision-making in finance: An overview and assessment of selected research. *Spanish Journal of Finance and Accounting/Revista Española de Financiación y Contabilidad*, 42(157):99–118.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Desmet, B. and Hoste, V. (2013). Emotion detection in suicide notes. *Expert Systems with Applications*, 40(16):6351 – 6358.

Dhall, A., Goecke, R., Joshi, J., Wagner, M., and Gedeon, T. (2013). Emotion recognition in the wild challenge 2013. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 509–516. ACM.

Dice, L. R. (1945). Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302.

Du, S., Tao, Y., and Martinez, A. M. (2014). Compound facial expressions of emotion. *Proceedings of the National Academy of Sciences*, 111(15):E1454–E1462.

Ekman, P. (1992). An argument for basic emotions. *Cognition & Emotion*, 6(3-4).

Facebook (2016). Reactions now available globally. Accessed: 2016-03-05.

Farías, D. I. H., Patti, V., and Rosso, P. (2016). Irony detection in twitter: The role of affective content. *ACM Transactions on Internet Technology (TOIT)*, pages 19:1–19:24.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E. H., and Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1606–1615.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., and Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *EMNLP*.

Filatova, E. (2012). Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *LREC*, pages 392–398.

Filik, R., Hunter, C. M., and Leuthold, H. (2015). When language gets emotional: Irony and the embodiment of affect in discourse. *Acta psychologica*, 156:114–125.

Filik, R., Turcan, A., Thompson, D., Harvey, N., Davies, H., and Turner, A. (2016). Sarcasm and emoticons: Comprehension and emotional impact. *The Quarterly Journal of Experimental Psychology*, 69(11):2130–2146. PMID: 26513274.

Fried, D. and Duh, K. (2014). Incorporating both distributional and relational semantics in word representations. *arXiv preprint arXiv:1412.4369*.

Gao, J., Zhou, M., Nie, J.-Y., He, H., and Chen, W. (2002). Resolving query translation ambiguity using a decaying co-occurrence model and syntactic dependence relations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 183–190, New York, NY, USA. ACM.

Gers, F. A., Schmidhuber, J. A., and Cummins, F. A. (2000). Learning to forget: Continual prediction with lstm. *Neural Computation*, 12(10):2451–2471.

Ghazi, D., Inkpen, D., and Szpakowicz, S. (2010). Hierarchical approach to emotion recognition and classification in texts. In *Proceedings of the 23rd Canadian Confer-*

ence on Advances in Artificial Intelligence, AI'10, pages 40–50, Berlin, Heidelberg. Springer-Verlag.

Ghazi, D., Inkpen, D., and Szpakowicz, S. (2014). Prior and contextual emotion of words in sentential context. *Computer Speech and Language*, 28(1):76 – 92.

Ghosh, A. and Veale, T. (2016). Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@NAACL-HLT 2016*, pages 161–169.

Ghosh, D., Guo, W., and Muresan, S. (2015). Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In *EMNLP*.

Giatsoglou, M., Vozalis, M. G., Diamantaras, K., Vakali, A., Sarigiannidis, G., and Chatzisavvas, K. C. (2017). Sentiment analysis leveraging emotions and word embeddings. *Expert Systems with Applications*, 69:214 – 224.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. *Processing*, pages 1–6.

González-Ibáñez, R., Muresan, S., and Wacholder, N. (2011). Identifying sarcasm in twitter: A closer look. In *Human Language Technologies: Short Papers*. ACL.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Gupta, N., Gilbert, M., and Fabbrizio, G. D. (2013). Emotion detection in email customer care. *Computational Intelligence*, 29(3):489–505.

Han, G. (2003). *Investigation of irony in modern Korean language*. Seoul: Yeoglag.

Hartmann, K., Siegert, I., Philippou-HĆ1/4bner, D., and Wendemuth, A. (2013). Emotion detection in hci: From speech features to emotion space. In *Analysis, Design, and Evaluation of Human-Machine Systems*, volume 12, pages 288–295.

Hernández-Farías, I., Benedí, J.-M., and Rosso, P. (2015). *Applying Basic Features from Sentiment Analysis for Automatic Irony Detection*, pages 337–344. Springer.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hong, Y., Lu, J., Yao, J., Zhu, Q., and Zhou, G. (2012). What reviews are satisfactory: Novel features for automatic helpfulness voting. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 495–504, New York, NY, USA. ACM.

Huang, C.-F. and Lin, E.-J. (2013). An emotion-based method to perform algorithmic composition. In *Proceedings of the 3rd International Conference on Music & Emotion (ICME3), Jyväskylä, Finland, 11th-15th June 2013. Geoff Luck & Olivier Brabant (Eds.). ISBN 978-951-39-5250-1.* University of Jyväskylä, Department of Music.

Izard, C. E. (1971). *The Face of Emotion*. Appleton-Century-Crofts, New York.

Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50.

Jorgensen, J. (1996). The functions of sarcastic irony in speech. *Journal of Pragmatics*, 26(5):613–634.

Joshi, A., Agrawal, S., Bhattacharyya, P., and Carman, M. (2017). Expect the unexpected: Harnessing sentence completion for sarcasm detection. *arXiv preprint arXiv:1707.06151*.

Joshi, A., Sharma, V., and Bhattacharyya, P. (2015). Harnessing context incongruity for sarcasm detection. In *ACL and IJCNLP, 2015, Short Papers*, pages 757–762.

Joshi, A., Tripathi, V., Bhattacharyya, P., and Carman, M. (2016a). Harnessing sequence labeling for sarcasm detection in dialogue from tv series friends. *CoNLL 2016*, page 146.

Joshi, A., Tripathi, V., Patel, K., Bhattacharyya, P., and Carman, M. J. (2016b). Are word embedding-based features useful for sarcasm detection? In *EMNLP*.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland. Association for Computational Linguistics.

Katz, P., Singleton, M., and Wicentowski, R. (2007). Swat-mp:the semeval-2007 systems for task 5 and task 14. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 308–313, Prague, Czech Republic. Association for Computational Linguistics.

Khattri, A., Joshi, A., Bhattacharyya, P., and Carman, M. J. (2015). Your sentiment precedes you: Using an authors historical tweets to predict sarcasm. In *WASSA 2015*, page 25.

Khokhlova, M., Patti, V., and Rosso, P. (2016). Distinguishing between irony and sarcasm in social media texts: Linguistic observations. In *Intelligence, Social Media and Web (ISMW FRUCT)*.

Kim, S.-M., Pantel, P., Chklovski, T., and Pennacchiotti, M. (2006). Automatically assessing review helpfulness. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 423–430, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kozareva, Z., Navarro, B., Vazquez, S., and Montoyo, A. (2007). Ua-zbsa: A headline emotion classification through web information. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 334–337, Prague, Czech Republic. Association for Computational Linguistics.

Krcadinac, U., Pasquier, P., Jovanovic, J., and Devedzic, V. (2013). Synesketch: An open source library for sentence-based emotion recognition. *T. Affective Computing*, 4(3):312–325.

Kreuz, R. J. and Caucci, G. M. (2007). Lexical influences on the perception of sarcasm. In *Workshop on Computational Approaches to Figurative Language*.

Kreuz, R. J., Long, D. L., and Church, M. B. (1991). On being ironic: Pragmatic and mnemonic implications. *Metaphor and symbol*, 6(3):149–162.

Krishnamoorthy, S. (2015). Linguistic features for review helpfulness prediction. *Expert Systems with Applications*, 42(7):3751 – 3759.

Kumar, S. S., Kumar, M. A., and Soman, K. (2017). Sentiment analysis of tweets in malayalam using long short-term memory units and convolutional neural nets. In *International Conference on Mining Intelligence and Knowledge Exploration*, pages 320–334. Springer.

Labutov, I. and Lipson, H. (2013). Re-embedding words. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 489–493.

Larsen, I., Vinther-Jensen, T., Gade, A., Nielsen, J., and Vogel, A. (2016). Do i misconstrue?: Sarcasm detection, emotion recognition, and theory of mind in huntington disease. *Neuropsychology*, 30(2):181–189.

Lee, S. and Choeh, J. Y. (2014). Predicting the helpfulness of online reviews using multilayer perceptron neural networks. *Expert Systems with Applications*, 41(6):3041–3046.

Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc.

Liebrecht, C., Kunneman, F., and van den Bosch, A. (2013). The perfect solution for detecting sarcasm in tweets# not.

Liew, J. S. Y., Turtle, H. R., and Liddy, E. D. (2016). Emotweet-28: A fine-grained emotion corpus for sentiment analysis. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.*

Liu, H., Gao, Y., Lv, P., Li, M., Geng, S., Li, M., and Wang, H. (2017). Using argument-based features to predict and analyse review helpfulness. *arXiv preprint arXiv:1707.07279.*

Liu, J., Cao, Y., Lin, C.-Y., Huang, Y., and Zhou, M. (2007). Low-quality product review detection in opinion summarization. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 334–342. Poster paper.

Liu, Y., Huang, X., An, A., and Yu, X. (2008). Modeling and predicting the helpfulness of online reviews. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 443–452, Washington, DC, USA. IEEE Computer Society.

Lu, Y., Tsaparas, P., Ntoulas, A., and Polanyi, L. (2010). Exploiting social context for review quality prediction. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 691–700, New York, NY, USA. ACM.

Lukin, S. and Walker, M. (2013). Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. In *Proceedings of the Workshop on Language Analysis in Social Media.*

Ma, X. and Hovy, E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Mahmud, Q. I., Mohaimen, A., Islam, M. S., et al. (2017). A support vector machine mixed with statistical reasoning approach to predict movie success by analyzing public sentiments. In *Computer and Information Technology (ICCIT), 2017 20th International Conference of*, pages 1–6. IEEE.

Marco Baroni, Georgiana Dinu, G. K. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, 1:238–247.

Martin, L. and Pu, P. (2014). Prediction of helpful reviews using emotions extraction. In Brodley, C. E. and Stone, P., editors, *AAAI*, pages 1551–1557. AAAI Press.

McAuley, J., Targett, C., Shi, Q., and van den Hengel, A. (2015). Image-based recommendations on styles and substitutes. In *SIGIR*, pages 43–52. ACM.

Meo, P. D., Musial-Gabrys, K., Rosaci, D., Sarnè, G. M. L., and Aroyo, L. (2017). Using centrality measures to predict helpfulness-based reputation in trust networks. *ACM Transactions on Internet Technology*, 17(1):8:1–8:20.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Mikula, M., Gao, X., and Machová, K. (2017). Adapting sentiment analysis system from english to slovak. In *Computational Intelligence (SSCI), 2017 IEEE Symposium Series on*, pages 1–8. IEEE.

Mohammad, S. (2012). #emotional tweets. In **SEM 2012*, Montréal, Canada.

Mohammad, S. M. and Kiritchenko, S. (2015). Using hashtags to capture fine emotion categories from tweets. *Comput. Intell.*, 31(2):301–326.

Mohammad, S. M., Kiritchenko, S., and Martin, J. (2013). Identifying purpose behind electoral tweets. In *WISDOM*.

Mohammad, S. M. and Turney, P. D. (2010). Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, pages 26–34, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mohammad, S. M. and Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon. 29(3):436–465.

Mudambi, S. M. and Schuff, D. (2010). What makes a helpful online review? a study of customer reviews on amazon.com. *MIS Q.*, 34(1):185–200.

Neviarouskaya, A., Prendinger, H., and Ishizuka, M. (2007). Textual affect sensing for sociable and expressive online communication. In *Proceedings of the 2Nd International Conference on Affective Computing and Intelligent Interaction*, ACII, pages 218–229, Berlin, Heidelberg. Springer-Verlag.

Neviarouskaya, A., Prendinger, H., and Ishizuka, M. (2009). Compositionality principle in recognition of fine-grained emotions from text. In Adar, E., Hurst, M., Finin, T., Glance, N. S., Nicolov, N., and Tseng, B. L., editors, *ICWSM*. The AAAI Press.

Neviarouskaya, A., Prendinger, H., and Ishizuka, M. (2010). Am: Textual attitude analysis model. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, pages 80–88, Stroudsburg, PA, USA. Association for Computational Linguistics.

Neviarouskaya, A., Prendinger, H., and Ishizuka, M. (2011). Affect analysis model: Novel rule-based approach to affect sensing from text. *Nat. Lang. Eng.*, 17(1):95–135.

Neviarouskaya, A., Prendinger, H., and Ishizuka, M. (2013). Attitude sensing in text based on a compositional linguistic approach. *Computational Intelligence*, pages n/a–n/a.

Nguyen, X.-D., Nguyen, M.-D., Tran, M.-V., Phan, X.-H., and Pham, S. B. (2017). Vnu-smm: A social media monitoring framework on vietnamese online news. In *Information and Computer Science, 2017 4th NAFOSTED Conference on*, pages 269–274. IEEE.

Niraula, N. B., Gautam, D., Banjade, R., Maharjan, N., and Rus, V. (2015). Combining word representations for measuring word relatedness and similarity. In *The Twenty-Eighth International Flairs Conference*.

O'Mahony, M. P. and Smyth, B. (2009). Learning to recommend helpful hotel reviews. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 305–308, New York, NY, USA. ACM.

O'Mahony, M. P. and Smyth, B. (2010). Using readability tests to predict helpful product reviews. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*, RIAO '10, pages 164–167, Paris, France, France.

Oraby, S., Harrison, V., Reed, L., Hernandez, E., Riloff, E., and Walker, M. (2016). Creating and characterizing a diverse corpus of sarcasm in dialogue.

Özbal, G. and Pighin, D. (2013). Evaluating the impact of syntax and semantics on emotion recognition from text. In *Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing - Volume 2*, CICLing'13, pages 161–173, Berlin, Heidelberg. Springer-Verlag.

Parrott, Gerrord, W. (2001). *Emotions in Social Psychology*. Psychology Press, Philadelphia.

Pennebaker, J. W., Francis, M. E., and Booth, R. J. (2001). Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*.

Phillips, L. H., Allen, R., Bull, R., Hering, A., Kliegel, M., and Channon, S. (2015). Older adults have difficulty in decoding sarcasm. *Developmental psychology*, 51(12):1840.

Plutchik, R. (1980). *A general psychoevolutionary theory of emotion*, pages 3–33. Academic press, New York.

Plutchik, R. (2001). The nature of emotions. *American Scientist*, 89(4):344–350.

Pool, C. and Nissim, M. (2016). Distant supervision for emotion detection using facebook reactions. *arXiv preprint arXiv:1611.02988*.

Poria, S., Cambria, E., Hazarika, D., and Vij, P. (2016). A deeper look into sarcastic tweets using deep convolutional neural networks. In *COLING 2016, Osaka, Japan*.

Punyakanok, V. and Roth, D. (2001). The use of classifiers in sequential inference. In *NIPS*, pages 995–1001. MIT Press.

Purver, M. and Battersby, S. (2012). Experimenting with distant supervision for emotion classification. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 482–491, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rabiner, L. and Juang, B. (1986). An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16.

Rajadesingan, A., Zafarani, R., and Liu, H. (2015). Sarcasm detection on twitter: A behavioral modeling approach. In *WSDM '15*.

Rajaraman, A. and Ullman, J. D. (2012). *Mining of massive datasets*. Cambridge University Press, Cambridge.

Ren, Y., Wang, R., and Ji, D. (2016). A topic-enhanced word embedding for twitter sentiment classification. *Information Sciences*, 369:188 – 198.

Reyes, A., Rosso, P., and Buscaldi, D. (2012). From humor recognition to irony detection: The figurative language of social media. *Data Knowledge Engineering*.

Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., and Huang, R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. In *EMNLP*.

Riviello, M. T. and Esposito, A. (2016). Results for hungarian participants. In *On the Perception of Dynamic Emotional Expressions: A Cross-cultural Comparison*, pages 29–32. Springer.

Sahlgren, M. (2006). *The Word-space model*. PhD thesis, University of Stockholm (Sweden).

Seol, Y. S., Kim, D. J., and Kim, H. W. (2008). Emotion Recognition from Text Using Knowledge-based ANN.

Shrum, L., Liu, M., Nespoli, M., and Lowrey, T. M. (2013). Persuasion in the marketplace: How theories of persuasion apply to marketing and advertising. *The Sage Handbook of Persuasion: Developments in Theory and Practice*, pages 314–330.

Siegert, I., Hartmann, K., Glüge, S., and Wendemuth, A. (2013). Modelling of emotional development within human-computer-interaction.

Smith, P. and Lee, M. G. (2013). A ccg-based approach to fine-grained sentiment analysis in microtext. In *AAAI Spring Symposium: Analyzing Microtext*, volume SS-13-01 of *AAAI Technical Report*. AAAI.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., and Manning, C. D. (2011). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 151–161.

Sønderby, S. K. and Winther, O. (2014). Protein secondary structure prediction with long short term memory networks. *arXiv preprint arXiv:1412.7828*.

Staiano, J. and Guerini, M. (2014). Depechemood: a lexicon for emotion analysis from crowd-annotated news. *CoRR*, abs/1405.1605.

Strapparava, C. and Mihalcea, R. (2008). Learning to identify emotions in text. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, pages 1556–1560, New York, NY, USA. ACM.

Strapparava, C. and Valitutti, A. (2004). WordNet-Affect: An affective extension of WordNet. In *LREC*, pages 1083–1086.

Subasic, P. and Huettner, A. (2001). Affect analysis of text using fuzzy semantic typing. *Transactions on Fuzzy Systems*, 9(4):483–496.

Sulis, E., Irazú Hernández Farías, D., Rosso, P., Patti, V., and Ruffo, G. (2016). Figurative messages and affect in twitter. *Knowledge Based Systems*, pages 132–143.

Sun, Y., Li, Z. P., and Xia, Y. W. (2013). Emotional interaction agents in intelligent tutoring systems. *Applied Mechanics and Materials*, 347:2682–2687.

Suttles, J. and Ide, N. (2013). *Distant Supervision for Emotion Classification with Discrete Binary Values*, pages 121–136. Springer Berlin Heidelberg, Berlin, Heidelberg.

Tang, D., Wei, F., Qin, B., Yang, N., Liu, T., and Zhou, M. (2016). Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):496–509.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL*.

Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

Tocoglu, M. A. and Alpkocak, A. (2018). Tremo: A dataset for emotion analysis in turkish. *Journal of Information Science*, page 0165551518761014.

Trilla, T. and Alias, F. (2013). Sentence-based sentiment analysis for expressive text-to-speech. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(2):223–233.

Tsur, O., Davidov, D., and Rappoport, A. (2010). Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *ICWSM*.

Tsur, O. and Rappoport, A. (2009). Revrank: A fully unsupervised algorithm for selecting the most helpful book reviews. In *ICWSM*.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA. Association for Computational Linguistics.

Turney, P. D. (2001). Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the 12th European Conference on Machine Learning*, EMCL '01, pages 491–502, London. Springer-Verlag.

Tversky, A. (1977). Features of similarity. *Psychological Review*, 84:327–352.

Valitutti, R. (2004). Wordnet-affect: an affective extension of wordnet. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*.

van der Maaten, L. and Hinton, G. E. (2008). Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Wallace, B., Choe, D., Kertz, L., and Charniak, E. (2014). *Humans require context to infer ironic intent (so computers probably do, too)*, volume 2, pages 512–516. ACL.

Wallace, B. C. (2015). Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment. ACL.

Wang, W., Chen, L., Thirunarayan, K., and Sheth, A. P. (2012). Harnessing twitter "big data" for automatic emotion identification. In *SocialCom/PASSAT*, pages 587–592. IEEE.

Wang, Z., Wu, Z., Wang, R., and Ren, Y. (2015). Twitter sarcasm detection exploiting a context based model. In *International Conference on Web Information Systems Engineering*, pages 77–91. Springer.

Whissell, C. (2009). Using the revised dictionary of affect in language to quantify the emotional undertones of samples of natural language. *Psychological reports*, 105(2):509–521.

Wiggins, J. B., Grafsgaard, J. F., Boyer, K. E., Wiebe, E. N., and Lester, J. C. (2014). The relationship between task difficulty and emotion in online computer programming tutoring. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 721–721. ACM.

Xu, C., Bai, Y., Bian, J., Gao, B., Wang, G., Liu, X., and Liu, T.-Y. (2014). Rc-net: A general framework for incorporating knowledge into word representations. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 1219–1228, New York, NY, USA. ACM.

Yang, C., Lin, K. H.-Y., and Chen, H.-H. (2007). Building emotion lexicon from weblog corpora. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 133–136. Association for Computational Linguistics.

Yang, Y., Chen, C., and Bao, F. S. (2016). Aspect-based helpfulness prediction for online product reviews. In *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*, pages 836–843. IEEE.

Yang, Y., Yan, Y., Qiu, M., and Bao, F. S. (2015). Semantic analysis and helpfulness prediction of text for online product reviews. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 38–44.

Yao, Z., Sun, Y., Ding, W., Rao, N., and Xiong, H. (2018). Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 673–681, New York, NY, USA. ACM.

Yu, M. and Dredze, M. (2014). Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 545–550.

Zhang, M., Zhang, Y., and Fu, G. (2016). Tweet sarcasm detection using deep neural network. In *COLING*.

Zhang, R., Gao, Y., Yu, W., Chao, P., Yang, X., Gao, M., and Zhou, A. (2015). *Review Comment Analysis for Predicting Ratings*, pages 247–259. Springer International Publishing, Cham.

Zhang, Z. and Varadarajan, B. (2006). Utility scoring of product reviews. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pages 51–57, New York, NY, USA. ACM.

Zheng, K., Li, A., and Farzan, R. (2018). Exploration of online health support groups through the lens of sentiment analysis. In *International Conference on Information*, pages 145–151. Springer.