GuessMyMood: Text-based Emotion Classification using two-pass Neural Network Architecture

Laphonchai Jirachuphun, Varot Premtoon, Navi Tansaraviput

Abstract—Natural language sentences, especially in fictional narratives, often convey emotions that can be understood by human readers. However, unlike in speech, written text does not have pitch or other non-verbal cues that signal emotions. In order to extract the emotional elements of a sentence, a more representative set of features and models are required. In this paper, we explore the text-based emotion prediction problem by investigating effectiveness of different feature sets and existing models. Finally, we construct and evaluate a new two-pass computational model based on neural networks. Our feature extraction method uses an emotion lexicon and a dependency tree to form sentence feature vectors that are then fed into a neural network model to extract the sentence emotion in the first pass. The predicted emotions are then fed into a Long Short-Term Memory neural network (LSTM) for re-evaluation. We base our experiments on a fairy tale corpus containing 176 annotated stories with the total of 15,302 sentences. The proposed feature set, when using a one-pass feedforward neural network for training, yields the best averaged F1 and is better than our baseline. The two-pass neural networks model does not improve the results, potentially due to a shortage of training data and unbalanced emotion classes.

Index Terms—Text-based emotion prediction, Natural Language Processing, Neural Network.

1 Introduction

Automatic emotion classification in text is becoming increasingly important from an application point of views. Knowing humans emotional state, computers can respond to human more appropriately, which yields a better human-computer coordination [1]. The solution to such a problem will provide a significant improvement to text-to-speech systems, human-computer interaction, and opinion mining.

In texts, unlike in speech where vocal cues are present, emotions are not explicitly expressed. Effective communication involves not only information but also affective components. The ability to integrate emotions into text helps listeners better understand and appreciate the contents. Apart from making the story more interesting, emotional storytelling is also a useful therapeutic application for children with communication disorders [2].

The application of our work can further be applied to improve the automated emotion prediction system for user satisfaction evaluation and real-time customer question answering system. Nowadays, the real-time automated system is becoming increasingly popular. Adding emotional intelligence to chat interfaces provides more subtle emotional interpretation for the automated machine to better formulate the responses.

The common goal of all the studies in this area is developing a system that can detect emotions of the users and express various types of emotions from textual contents. There are two major challenges to this problem. The first challenge is that written texts lack major components of emotion appeared in audio such as pitch, tone and stress. Without these vocal indicators, words, when used in different contexts, can convey different emotions. The second challenge is the ambiguous nature of emotions. Evaluating one annotator against another yields only about 46% F1-score.

Ambiguity in interpreting emotions arises when the emotional statement is not specific or can be interpreted in multiple senses depending on the context. Consider an example, I am going to Florida next week. This statement may entail happy emotion if the speaker mentioned earlier with excitement that he is looking forward to travelling to Florida. However, this sentence can simply be a neutral declarative sentence. The sentence can even convey sad emotion if the subject did not chose to leave on his own accord. To resolve the ambiguity, a shared world or shared knowledge are required and the interpretation is carried out using the context. Automatic resolution of all these ambiguities contains several long-standing problems that make emotion classification task complex and challenging.

To address this, we propose a two-pass architecture based on neural network and Long Short-Term Memory neural network (LSTM). The architecture takes in input phrases of any length and predicts emotions at the sentence-level. The input phrases are partitioned into sentences. Each sentence is represented using a fixed-length feature vector. A sentence vector is parsed by the first neural network to generate initial emotion tag for each sentence. Neural networks perform well at learning which features in the feature set are important without human having to manually select the best subset of features. The predicted tags are then fed into an LSTM for re-evaluation.

In this paper, we aim to recognize the six emotions

• *Department of Electrical Engineering and Computer Science, Masschusetts Institute of Technology, Cambridge, MA, 02139.*

Manuscript received December 14, 2015

suggested by Ekman: happiness, sadness, anger, fear, disgust and surprise [3]. The rest of the paper is organized as follows. Section 2 summarizes related works, whereas section 3 explains dataset we use for training and testing our model. Next, section 4 describes our methodology for both feature selection and models we use for this problem. Section 5 presents our experiments and achieved results and section 6 concludes our paper and suggests future work.

2 Related Work

The notion of emotions in this paper is adopted from Ekmans emotional model (1993). According to Ekmans model, basic emotions comprise six categories: ANGER, DISGUST, FEAR, HAPPINESS, SADNESS and SURPRISE [3], which are universally recognized across all cultures.

Most studies focus their works on sentiment analysis, a study of classifying each sentence from a text as having positive, negative or neutral sentiment. Socher introduces Recursive Neural Tensor Network (RNTN) as a new classification model [4] RNTN is similar to Matrix-Vector RNN (MV-RNN) in which they both use compositional function to find relation between words in longer phase, but RNTN uses a much more powerful composition function, which performs better and faster.

Working on the similar problem, Sun instead used multilayer of Restricted Bolzmann Machine (RBM), a type of Convolution Neural Network, as a classification model for Chinese Microblog sentiment classification [6]. For feature extraction, each word he uses both emotional information such as parts of speech (POS) tagging and semantic information such as degree words (seldom, often, extremely, etc.) as word-level feature. He also proposes ConCAE method to extract context information from the blog and use them as sentence-level feature.

Another work by Alm investigates the emotion classification problem using SNoW learning architecture [7]. SNoW is a sparse network of linear functions over a pre-defined or incrementally learned feature space. In the paper, the architecture is tested in a set of 30 predefined features, which contributes to the improvement in classification results.

In general, more complex dynamic of interpretation of the contents is achieved after rereading. Based on the preceding concept, we architect our system to perform two-pass classification. The second pass re-evaluates the emotions results from the first pass taking into account the emotions from nearby sentences.

3 Dataset

3.1 Corpus characteristic

Supervised machine learning approaches for automated emotional classification task require labelled dataset. We utilize the annotated a fairy tale corpus from [EBBA CECILIA OVESDOTTER ALM]. Alm annotated 176 children stories of 15,302 sentences in total, using seven tags to represent Ekmans six basic emotions and neutral. If the emotion is a SURPRISE, the surprise tag will be classified into one of two categories: positively surprised, and negatively surprised. However, for our experiments, we decide to merge both positive and negative labels of surprise into one category: SURPRISE.

Fig. 1. Overview

Subcorpus	Number of Stories	Number of Sentences
B. Potter	19	1946
Grimm	80	5360
H.C. Andersen	77	7996
Total	176	15302

The corpus is annotated at sentence-level in three sets of children's stories: Beatrix Potter, H. C. Andersen, and the Brothers Grimm by six trained annotators. For each sentence in the corpus, two selected annotators assign primary emotions and moods to the given sentence. Primary emotion describes the emotion of the subject, i.e. the feeler of the emotion, such as the emotion of the speaker for a given sentence, or the emotion of character acting in a given context, whereas mood describes the general feeling for the context of the sentence.

We find that the two annotators disagree rather frequently. One annotator seems to be more conservative and tags more sentences as neutral. We also find the notation of mood to be fuzzier with more frequent occurrences of disagreement between two annotators. Therefore, we decided to use the primary emotion tags as annotated by the first annotator to develop the dataset for our model. The second annotator will be evaluated against the first annotator to illustrate how much humans agree or disagree when it comes to emotion classification.

Neutral	Happy	Surprised	Sad
66.3%	10.5%	5.4%	5.4%

Angry	Fearful	Disgusted
4.8%	4.6%	3.0%

The categories of emotion in the dataset are not equally represented. The majority of the sentences in the corpus are annotated as neutral. This poses the problem of imbalance in emotion classes, as will be discussed later in section 4.2.

4 Methodology

We divide the dataset into three sets: training set, development set, and test set. For each set, we first extract a feature vector of each sentence. The sentence feature vectors are fed into the classification models for training. The trained models predict emotion tags on the test sentences. Finally, we evaluate precision, recall, and F1 score of the predicted tags.

4.1 Feature Selection

Previous studies show that certain characteristics and clues, such as punctuation, parts of speech, syntax, and lexicon play an important role in expressing emotions in textual contents. Sentences with exclamation points are usually associated with ANGER or SURPRISE emotions, while sentences towards the end of a story frequently have either HAPPINESS or SADNESS emotions. Moreover, generally

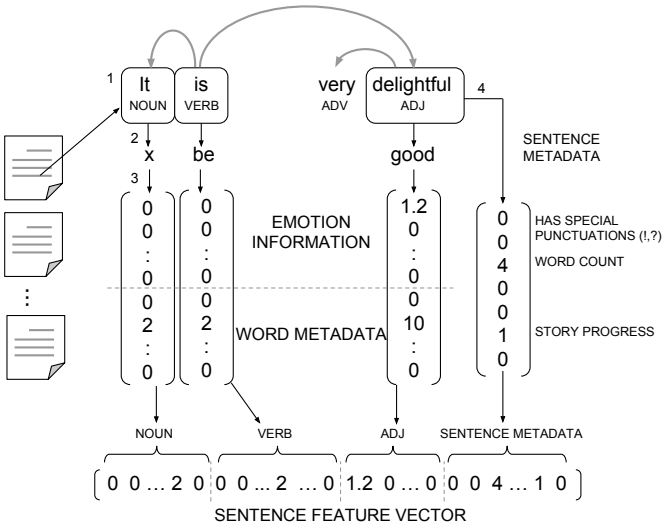

Fig. 2. Sentence
Word emotion features and word metadata features compose word-level feature vectors. These vectors, and sentence metadata feature vector, are concatenated to form sentence feature vectors.

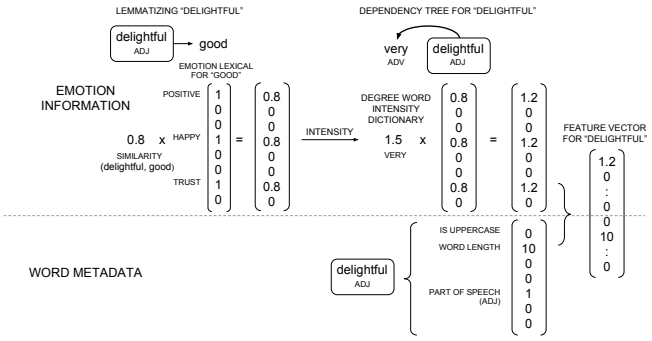

Fig. 3. Word
Emotion feature vector is constructed by taking emotional association vector from NRC Emotion Lexicon, scaled by cosine similarity and degree word factors. Emotion vector and word metadata feature compose word-level feature vector.

adjectives and verbs are the words that most entail the emotion of a sentence, so we pay more attention to these words. For each sentence, we select features according to these characteristics, and we separate them into three parts: emotion information, word metadata, and sentence metadata features.

4.1.1 Emotion Information Features

In this part, we focus mainly on important words that bring about the emotion of a sentence. We use NRC Emotion Lexicon, which is a dictionary that maps English words to emotional associations, to construct the words emotional association vectors. We also look at degree words, such as very or rarely, that modify the emotion words to adjust the emotion intensity. Finally, we concatenate all the vectors into one feature for the whole sentence. More details on the extraction are as follows:

1) First we select emotion words from a sentence using their parts of speech (POS) taggings. In our model, we select five words from the sentence: one noun, two adjectives, and two verbs. This POS selection yields the highest F1 score in development set. Note that if the sentence does not have enough adjectives or verbs, we will use zero vector as their representatives. (Figure X)
2) For each selected word, we look for the most similar word in the NRC Emotion Lexicon (EmoLex). Similarity is measured by cosine similarity of the word embeddings, which we obtain from spaCy, a python NLP library. EmoLex maps each English word to a binary vector, where each element represents whether the word is associated with one of the eight emotions: anger, anticipation, disgust, fear, joy, sadness, surprise, and trust, or one of the two sentiments: positive and negative. We therefore take that binary vector and multiply it with the cosine similarity score to create the feature vector of our word. (Top Left of Figure Y)
3) Because an emotion word may also have modifiers, such as very or rarely, we need to scale the emotion feature vector according to these modifiers. We call these modifiers degree words. We created a list of what we determined as degree words, and hand annotate each word with a real value that should correspond to its degree factor. For example very is assigned with value 1.5. This is what we call the Degree Word Dictionary. For each emotion word selected in step 1, we obtain the list of its modifiers by looking at the sentences dependency tree, and then multiply the emotion vector from step 2 with the degree values of the modifiers according to the Degree Word Dictionary. Words that do not appear in the dictionary has the degree value 1. (Top Right of Figure Y)
4) Finally, the emotion information feature is the concatenation all emotion feature vectors of the selected five emotion words.

4.1.2 Word Metadata Features

For this part of the feature, we create word metadata feature for each emotion word we selected in the first step of extracting emotion information feature. The feature set is as follows:

1) Completely upper-case words
2) Word length in letters
3) Parts of speech tagging

We expect words that are all capitalized to represent strong emotional state. We also suspect the correlation between emotions and the length of words in letters; hence, we append the length of the word into our word metadata feature. Additionally, due to the fact that some parts of speech are more likely to represent emotions than others, we encode parts of speech attribute into word metadata feature vector. Parts of speech are represented by a fix-length subarray.

4.1.3 Sentence Metadata Features

Similar to word metadata feature, sentence metadata feature is a special characteristic of the whole sentence. Below is the feature set we use:

1) Whether sentence is the first sentence in the document
2) Whether sentence is the last sentence in the document
3) Special punctuation (!, ?)
4) Sentence length in words
5) Ranges of the story progress (0%-20%, 20%-40%, 40%-60%, 60%-80%, or 80%-100% of the document)
6) Word count for adjectives, adverbs, verbs, and nouns.

4.2 Models

After constructing sentence feature vectors, we perform multiclass classification using the following two-pass architecture. We divide the training set into two sets of roughly equal number of documents, called Training Set 1 and Training Set 2. Training Set 1, along with its emotion tags, are rebalanced. Rebalancing is done by repeating the sentences whose tags are less frequent so that, at the end, every tag has the same number of instances. We also reorder the sentences so that the tags appear in the circular pattern: angry, disgusted, fearful, neutral, happy, sad, surprise, and then back to angry, disgusted, fearful, and so on. We do this to avoid certain tags (such as neutral) to dominate the training set. The rebalanced sentence vectors are entered into a feedforward neural network (ffNN) for training. The network consists of two hidden layers, each with 128 hidden units with tanh activation function. The output layer is a softmax layer with seven units corresponding to the seven emotions. Between the two hidden layers and between the last hidden layer and the output layer, we also added a dropout layer to reduce overfitting [5].

To train the second pass model, we use the trained ffNN to predict the tags of the sentences in Training Set 2. Afterward, we augment each feature vector in Training Set 2 with its predicted tag as well as the predicted tags of the two sentences before and two sentences after it. Therefore, each of the modified feature vectors in Training Set 2 will include the predicted tags of five nearby sentences. We do this to encode more context into the feature vector. We do not rebalance or reorder Training Set 2 as we wish to preserve the order and context of the sentences. We feed the modified feature vectors into an LSTM classifier for training. The LSTM layer has 128 output units. We then added a softmax layer with seven nodes to output the emotion tag probabilities. We use Categorical Cross-entropy as the loss function.

To evaluate our model, we extract feature vectors from the test set. We get first-pass prediction from our ffNN model, augment each sentence feature vector with the predicted tags as we do in training, and then get second-pass prediction from the LSTM model. The second-pass predicted tags are evaluated against the gold standard tags we obtain from the corpus.

As alternatives, we try using standard models, including One-Versus-The-Rest SVM and decision tree. We also try using only one layer of ffNN (without LSTM), as well as a two-pass model where the second layer is also an ffNN rather than an LSTM. Unlike LSTM, none of these alternative models is a recurrent model in which the input sentence order matters. Therefore, we perform rebalancing on the training data for all of the models.

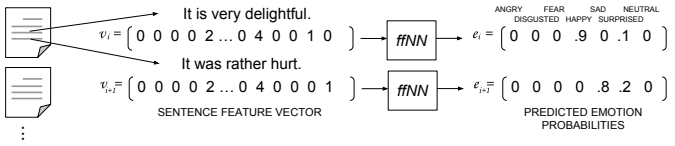

Fig. 4. First-pass model

First-pass model is a feedforward neural network model, which takes feature vector and predicts emotion probability vectors for the sentences.

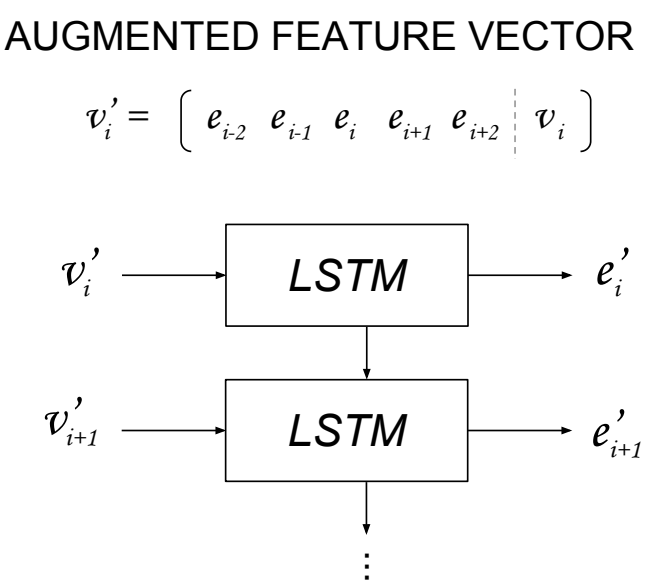

Fig. 5. Second-pass model

Second-pass model is an LSTM model, which takes the augmented feature vector and predicts emotion classes.

5 Experiments

We calculate the precision and recall given by the second human annotators and implement two baseline models to estimate accuracy bounds. Our baseline models include a classifier that outputs uniformly random emotion for each sentence and a classifier that always outputs neutral regardless of the input sentences. The results are shown in [table 5.1]. Note that the metrics we use are the *macro average* precision, recall, and F-measure, which are the *unweighted means* of the precision, recall, and F-measure scores of individual emotion classes. We choose the unweighted averages so that the models that output too many neutral tags will not receive high scores. We also wish to treat all emotions as equally important.

Baselines	Precision	Recall	F1
2nd human annotator	58.2%	40.3%	45.6%
random guess	14.3%	13.4%	9.8%
all neutral	9.5%	14.3%	11.4%

[Table 5.1] describes the computed *macro average* precision, recall, and F-measure for second human annotator and the two baselines.

The F1 score of the human annotator at 45.6

The corpus is splitted into training (65%), development (15%), and test (20%) sets. For all models, we use the development set to determine the optimal hyperparameters, and we present the results from the test set. As mentioned in the Feature Extraction section, the optimal performance is achieved when we take into account one noun, two verbs, and two adjectives for each sentence with only special punctuation sentence metadata feature. The models we test include: decision tree; SVM; one-pass feedforward neural network, which is similar to the first-pass of the model proposed in section 4.2 without the LSTM second pass; two-pass feedforward neural network and LSTM, which is the model we propose in section 4.2; and two-pass feedforward neural network model, which is similar to the two-pass model we propose but with feedforward neural network for both passes. We also perform rebalancing on all models, except the second pass of the two-pass ffNN + LSTM model. (The reason is explained earlier in section 4.2.)

Model	Precision	Recall	F1
Decision Tree	17.9%	17.9%	17.8%
SVM	16%	16.2%	13.2%
1-pass ffNN	21.6%	25.8%	20.4%
2-pass ffNN + LSTM	16.6%	14.3%	11.5%
2-pass ffNN + ffNN	20.2%	23%	18%

[Table 5.2] The *macro average* precision, recall, and F-measure.

The experiments show that the one-pass feedforward neural network yields highest average F1 score of 20.4%, which is considerably higher than the baselines. The performance of the single neural network model outperforms both two-pass feedforward neural network with LSTM and two-pass feedforward neural network as well as other standard learning algorithms.

As shown in the results, the two-pass neural network models do not improve the results. The performance of feedforward neural network is inferior because of the lack in training data. In the two pass architectures, as we add another feedforward neural network on top of the single neural network model and transform the one-pass model into two-pass, the training set must be splitted into two sets to train both networks resulting in lower prediction performance. The two-pass neural networks model with LSTM suffers from the same problem at even a greater degree, as LSTM is more sophisticated and should require even more amount of data.

In addition to the shortage of the training set, the two-pass neural networks model with LSTM significantly underperforms other models potentially by the reason of the unbalanced dataset. Unlike other models, the dataset fed into the two-pass neural networks model with LSTM is not rebalanced because we hope that LSTM will capture the emotion transition and relation between nearby sentences, which may have a strong impact on the learning process of LSTM. Balancing the data will alter the sequence of sentences and emotions in the input data. As a results, LSTM suffers from the data imbalance and almost always output neutral, which explains the low macro average F1.

Model	Precision	Recall	F1
1-pass ffNN	23.1	15.1%	13.0%
2-pass ffNN + LSTM	9.5%	14.3%	11.4%
2-pass ffNN + ffNN	22.6%	15.4%	13.4%

[Table 5.3] displays the precision, recall, and F1-score for each model without rebalancing on all sentence features.

We also experiment on the models without rebalancing, as presented in Table 5.3. Both feedforward network without rebalancing models have significantly lower results than with rebalancing due to data imbalance problem. ffNN + LSTM remains roughly the same, but still lower than the feedforward neural network models because of the complexity of the model relative to the available training data.

6 Conclusion

Evaluating a human annotator revealed a difficult nature of emotion classification problem. We explore the two-pass feedforward neural network and the two-pass feedforward neural network with LSTM. The experiment shows that one-pass single neural network outperforms all other classification models with data rebalancing. The model performs significantly better than all baselines and other standard algorithms. The two-pass models did not improve the results, potentially due to complexity of the models relative to the available training. LSTM also suffers from the fact that we cannot rebalance the data. We do believe, however, that with more data we will perform better with the two-pass feedforward neural network model.

There are plenty of rooms for improvement on this task. First, we can use semi-supervised learning to exploit a larger corpus of unlabelled fairy tales by training three models on the labelled corpus and using tri-learning on the unlabelled corpus. This will make our complex model perform better and hopefully outperform the simple one-pass model. We can also improve on our feature set. For example, we may run sentiment analysis on the sentences and use the results as our feature. Sentiment should have a strong correlation with emotion and should improve our model significantly.

References

[1] R. W. Picard, E. Vyzas, and J. Healey (2001), *Toward Machine Emotional Intelligence: Analysis of Affective Physiological State*, IEEE Transactions Pattern Analysis and Machine Intelligence, Volume 23, No. 10, pp. 1175-1191, October 2001.

[2] van Santen, J. P. H., L. Black, G. Cohen, A. B. Kain, E. Klabbers, T. Mishra, J. de Villiers, and X. Niu (2003). *Applications of computer generated expressive speech for communication disorders*. In Proceedings of Eurospeech, pp. 16571660.

[3] P. Ekman (1994). *All emotions are basic*. In P. Ekman and R. J. Davidson (Eds.), The Nature of Emotion: Fundamental Questions, pp. 1519. Oxford: Oxford University Press.

[4] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D.Manning, A. Y. Ng, C. P. Potts. *Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank*. In Conference on Empirical Methods in Natural Language Processing, 2013b.

[5] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. Journal of Machine Learning Research 15, pages 1929-1958, 2014

[6] X. Sun, F. Gao, C. Li, and F. Ren. *Chinese Microblog Sentiment Classification Based on Convolution Neural Network with Content Extension Method.*. Proceedings of the Sixth International Conference on Affective Computing and Intelligent Interaction, pages 408-414. ACII, 2015

[7] A. Carlson, C. Cumby, N. Rizzolo, J. Rosen, and D. Roth. *The SNoW Learning Architecture.* . Technical Report UIUCDCS-R-99-2101, UIUC Comp. Sci. 1999.