Mixed Feelings: Natural Text Generation with Variable, Coexistent Affective Categories

Lee Kezar
Mathematics and Computer Science Department, Rhodes College
Memphis, Tennessee, USA
kezarlee@gmail.com

Abstract

Conversational agents, having the goal of natural language generation, must rely on language models which can integrate emotion into their responses. Recent projects outline models which can produce emotional sentences, but unlike human language, they tend to be restricted to one affective category out of a few (e.g. Zhao et al. (2018)). To my knowledge, none allow for the intentional coexistence of multiple emotions on the word or sentence level. Building on prior research which allows for variation in the intensity of a singular emotion (Ghosh et al., 2017), this research proposal outlines an LSTM (Long Short-Term Memory) language model which allows for variation in multiple emotions simultaneously.

1 Introduction

In closing her landmark paper on affective computing, Rosalind Picard charges researchers of artificial intelligence with a task. She writes, "Computers that will interact naturally and intelligently with humans need the ability to at least recognize and express affect" (Picard, 1995). While Picard herself has since spent her time primarily studying the physiology behind emotion and health, there is also a strong relationship between linguistics and emotion. The task for computer scientists who study this relationship is to symbolize and express emotion through verbal language alone.

Although this goal easy to articulate, accomplishing it has proven to be quite challenging. However, in the same way AI researchers acquired valuable insight from reviewing models of cellular neuroscience to produce an artificial neural network, perhaps we can demystify affect generation by reviewing psychological models which build on neuro-biological findings in regards to human emotion.

1.1 Need for Affective Mixing

I will now summarize two key ideas from these findings: emotion/language dynamic and classification of emotion.

First, emotion usually precedes language. In describing phenomenal consciousness as it pertains to emotion, Carroll Izard wrote that "an emotion feeling remains functional and motivational without being symbolized and made accessible in reflective consciousness via language" (Izard, 2009). One might support this claim by noting how complex emotional qualia feels and how rather limited language can be. For example, how accurate is it to say that "Alice is happy"? To what extent is she happy? Is it the same happiness that she feels when being in good company, or in favorable weather? Do they only differ in magnitude, or also along some other dimension? Or as a second example, can you recall a moment where you couldn't describe how you were feeling, but felt it nonetheless? Clearly, emotion is rather difficult to express in simple words. Yet, recent affective generation techniques tend to presume that they fall neatly into one of five or six discrete categories (Zhao et al., 2018). Recently, researchers added nuance to affect generation via variation in intensity (Ghosh et al., 2017), but to my knowledge no model adds nuance along extra dimensions, such as other emotions.

Second, emotion is usually the conflation of two distinct phenomena: "basic emotions" and "dynamic emotion-cognition interactions" (Izard, 2009). The basic emotions are linked to old evolutionary stimuli and are more automatic (e.g. fear

Proceedings of ACL 2018, Student Research Workshop, pages 141–145
Melbourne, Australia, July 15 - 20, 2018.

as a response mechanism to avoid danger). That is to say, basic emotions have little involvement with cognition. Emotion schemas, on the other hand, result directly from interactions between cognition and emotion. This type of emotion is underscored by research which casts emotion and cognition as interdependent processes (Storbeck and Clore, 2007), closely mirroring Picard's insistence that intelligence is comprised of emotion.

The categorizations usually found in lexicons (e.g. Linguistic Inquiry and Word Count or LIWC) treat words as stimuli which humans respond to with emotions. This model aligns closely with basic emotions, and supposes language precedes emotion. However, in actuality humans tend to incorporate cognitive processes such as memory and perspective-taking, allowing us to have multiple emotions simultaneously. If we are to use this as inspiration for generating affective text algorithmically, we might then permit the AI to intentionally express multiple emotions.

1.2 Project Overview

Having established this need for emotional intelligence in AI (including natural language processing), and reviewed psychological research in this area, we might conclude that emotions should not be modeled as singular, discrete categories but continuous and constantly mixing.

This project aims to create an algorithm which is capable of taking some priming text and the desired affective state which can vary along different emotional dimensions simultaneously, and produce a corresponding utterance. This algorithm is primarily meant for conversational agents, but could be adapted for other purposes.

1.3 Use Cases

Mixing emotions are not only more realistic to human emotion, since they allow for significantly more flexibility in expression, but also more helpful for specific applications. I will introduce three such applications here.

First, the ability to mix emotions in a continuous fashion allows for conversational agents to gradually and imperceptibly shift the tone of a conversation toward a new tone. Human tendency to mirror the emotions of others through empathy would make this an effective strategy to improve attitude and emotional outlook.

Second, mixing emotions in this way also allows for an extra layer of nuance in conversational practice, as we are affected by emotional language as often as we produce it. For example, in simulating a realistic conversation, the deliverance of emotionally-charged statements should cause the computer to respond appropriately. However, models which treat emotion as discrete categories would "overreact" and abruptly switch from one emotion to another.

Third, realistic personality can be introduced as a tendency to hover near or avoid specific points. Naturally, the conversational agent will vary in emotion, but ultimately return to some default state. For example, if an agent were to express optimistic personality, it might impose some minimum on the *joy* vector, and a maximum on the *sadness* and *anger* vectors. This is not possible with models that suppose emotions are discrete categories.

1.4 Algorithm Overview

We can encapsulate this goal with a broad formula

$$g(p, e) = w \tag{1}$$

where p is the priming text, e is a quadruple of values such that

$$e_x \in \mathbb{R} \mid x \in \{\text{j, s, f, a}\}, 0 \leq e_x \leq 1 \tag{2}$$

which correspond to the intensity of *joy, sadness, fear, anger*, respectively; and w is an array of words and punctuation which represent the algorithm's response to p with emotional state e. The emotional categories are selected to correspond with the DepecheMood database, which I will describe in section 3.2. The priming text can be a sentence fragment which the user seeks to complete or a natural sentence which the algorithm is meant to respond to.

The purpose for g is to be embedded into a conversational setting with improvements on parsing and production. I will not detail what this embedding looks like for sake of brevity and coherency.

An example set of sentences generated across a gradient beginning at high happiness ($e_j \approx 1$), low fear ($e_f \approx 0$) and ending at moderate happiness ($e_j \approx 0.5$), high fear ($e_f \approx 1$) might look like this:

- I am happy to go to work today
- I am content to go ...
- I am hesitant to go ...
- I am worried to go ...
- I am nervous to go ...
- I am anxious to go ...

since anxiety is, in a sense, the combination of fear and partial excitement or happiness.

2 Related Work

In the last two years, different models have been used to create conversational agents which can express affect. Namely, the Emotional Chatting Machine (ECM) (Zhao et al., 2018) and Affect-LM (Ghosh et al., 2017). These models are motivated by the psychological finding that agents with subtle expressivity can improve the affective state of the user (Prendinger et al., 2005). These projects utilize an array of emotional categories including *liking, happiness, sadness, disgust, anger*, and *anxiety*.

To accomplish this, different language models which utilize machine learning algorithms have been crafted and tested for accuracy and grammaticality. The most popular include feed-forward neural networks, recurrent neural networks (RNNs), and long short-term memory (LSTM) neural networks (Sundermeyer et al., 2015). Among these, the LSTM model is notably superior for establishing long-term dependencies in text, and reducing the vanishing gradient problem found in RNNs. This is the method used by (Ghosh et al., 2017), with an additional network for including emotionally-charged words of the desired strength. Importantly, they accommodated and tested for loss in grammaticality, which was predicted for expressions of intense emotion but not low emotions, where standard LSTMs typically suffice.

3 Model

3.1 LSTM Language Model

The LSTM Language Model allows for the use of all prior words as evidence in predicting the next best word, w_t. We can write this prediction as a probability like so:

$$p(w_1^M) = \prod_{i=1}^{M} p(w_i|w_1^{i-1}) \quad (3)$$

for a sequence of M words. We can utilize the LSTM as a function l of the prior words w_1^{i-1} to calculate $p(w_i)$. An additional bias term b_i corresponding to unigram occurrence of w_i may be included to favor more common words, as in Ghosh et al. (2017). The output layer of l summed with this bias term then pass through a softmax activation function to normalize the outputs, producing

$$p(w_i|w_1^{i-1}) = \text{softmax}(l(w_1^{i-1}) + b_i) \quad (4)$$

The algorithm would simply repeat this calculation, each step incrementing i and selecting the most probable word, until a period is produced indicating the end of the sentence and completion of the algorithm.

3.2 Incorporating Affective Data

The DepecheMood lexicon contains affective data for over 13,500 words, each rated along a continuous interval $[0, 1]$ for eight affective dimensions: $\{$*Fear, Amusement, Anger, Annoyance, Indifference, Happiness, Inspiration, Sadness*$\}$ (Staiano and Guerini, 2014). As a comical American example, DepecheMood rates the word "president" as $\{0.2, 0.346, 0.626, 1.0, 0.528, 0.341, 0.0, 0.115\}$ respectively – that is, moderately infuriating, never inspiring, and completely annoying.

For this project, I would use the more typical categories of *joy, sadness, fear*, and *anger*. This lexicon contrasts other popular lexicons with affective data, e.g. LIWC, in that these ratings are *continuous* along $[0, 1]$. This property allows for more precise matching to the affective state, and movement along a gradient between the four dimensions.

In addition to the bias term from the previous section, we would account for affect by introducing a third term $d(w_i, e)$ which favors words most affectively similar to the desired output.

Graphically, this equation maps the vocabulary V into four-dimensional space corresponding to each word's emotional valences, and calculates the distance between e, a point in this space, and each word $w_i \in V$. This inverse distance function can be written as such:

$$d(w_i, e) = \text{softmax}\left(\frac{1}{\sqrt{\sum_{j=1}^{|e|}(w_{ij} - e_j)^2}}\right) \quad (5)$$

where j enumerates each of the four affective dimensions, w_{ij} is the intensity along that dimension e_j is the target intensity.

3.3 Optimizing $d(w_i, e)$

In its current form, this function requires a calculation for each $w_i \in V$ which is has a subideal time complexity of O(n). We can reduce this to O($\log n$) by estimating e to its nearest neighbor in V (a O($\log n$) operation, as I will soon explain), whose distances to every other word can be precomputed and accessed via a hash table O(1). This replacement can be done without loss of accuracy due to the density of the data set.

To find e nearest neighbor in V, we can utilize a modified QuadTree algorithm to reduce the search space to some arbitrary n much smaller than 13,500. To briefly review, this algorithm organizes a set of data points into a hierarchical tree with leaf nodes containing a list of less than n points (Finkel and Bentley, 1974). Querying this tree to find nearby data points has been empirically shown to take on average O($\log n$) time, a modest improvement over O(n).

3.4 Avoiding Over-Emphasis

As Zhao et al. (2018) noted in their ECM project, "emotional responses are relatively short lived and involve changes." These changes, which the authors call "emotion dynamics", involve modeling emotions as quantities which decays at each step. This avoids the problem of over-emphasizing the input e by repeatedly expressing the same state, unintentionally compounding its strength. Their implementation of this decay involves updating e_j in (5) by subtracting w_{ij} for each dimension j. Therefore, upon completion, e would be close to $[0, 0, 0, 0]$.

Preliminary experimentation would certainly need to reveal the appropriate weights for g, d, and b_i such that precision of emotion does not sacrifice grammaticality.

4 Implementation, Training, and Review

For the LSTM, we would follow the suggestion of Sundermeyer et al. (2012) and implement a network using TensorFlow[1] with two hidden layers of 200 nodes: the first being a projection layer of standard neural network units and the second being hidden layer of LSTM units. The output layer would also have 200 nodes.

The same authors later suggest training the network using the cross-entropy error criterion, using the function

$$F(A) = -\sum_{i=1}^{M} \log p_A(w_i | w_{i-n+1}^{i-1}) \quad (6)$$

where M is the size of the training corpus (Sundermeyer et al., 2015). For a stochastic gradient descent algorithm, we can obtain a gradient using epochwise backpropogation through time (BPTT) on the first pass, and update the weights on the second pass as specified in Sundermeyer et al. (2014).

The training corpus for this algorithm would need only be some collection of natural dialogue. This can be catered to the environment it will be used in, but for our purposes the Ubuntu Dialogue Corpus will be used for its generality, accessibility, dyadic nature, and size (Lowe et al., 2016).

Additional funds have been secured to utilize Amazon's Mechanical Turk to analyze loss in grammaticality and verify the manipulation of e by simply assessing sentences along a Likert scale and comparing this data to the intended affect. I predict that certain categories and combinations will be harder to express in text than others, resulting in higher variability and weaker correlations. For example, *anger* and *sadness* (an approximation of *remorse*) may be rather easy to express, but *joy* and *sadness* may be difficult. Visualizing these strengths and weaknesses will be an interesting reflection of the English language.

5 Conclusion

In this research proposal, I have given a brief overview of a natural language generation algorithm which is capable of producing utterances expressive of multiple emotional dimensions simultaneously. The DepecheMood lexicon enables the

[1] http://www.tensorflow.org

mapping of words into n-dimensional space, allowing us to prefer words which minimize the distance between it and the target affective state along the four emotional dimensions.

After clarifying implementation details such as LSTM node construction, neural architecture, and use of the training corpus, this algorithm has the potential to add further nuance to our current models of generating affect which are consistent with psychological and neuro-biological findings.

References

R. A. Finkel and J. L. Bentley. 1974. Quad trees: A data structure for retrieval on composite keys. *Acta Informatica*, 4(1):1–9.

Sayan Ghosh, Mathieu Chollet, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. 2017. Affect-lm: A neural language model for customizable affective text generation. *ACL*, pages 634–642.

Carroll E. Izard. 2009. Emotion theory and research: Highlights, unanswered questions, and emerging issues. *Annual Review of Psychology*, 60:1–25.

Ryan Lowe, Nissan Pow, Iulian V. Serban, and Joelle Pineau. 2016. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems.

Rosalind W. Picard. 1995. Affective computing. Technical report, Massachusetts Institute of Technology, Perceptual Computing Section.

Helmut Prendinger, Junichiro Mori, and Mitsuru Ishizuka. 2005. Using human physiology to evaluate subtle expressivity of a virtual quizmaster in a mathematical game. *Human-Computer Studies*, 2:231–245.

Jacopo Staiano and Marco Guerini. 2014. Depechemood: a lexicon for emotion analysis from crowd-annotated news.

Justin Storbeck and Gerald L. Clore. 2007. On the interdependence of cognition and emotion. *Cognitive Emotion*, 21(6):1212–1237.

Martin Sundermeyer, Hermann Ney, and Ralf Schlter. 2012. Lstm neural networks for language modeling. In *Interspeech*.

Martin Sundermeyer, Hermann Ney, and Ralf Schlter. 2014. Rwthlm-the rwth aachen university neural network language modeling toolkit. In *Interspeech*, pages 2093–2097.

Martin Sundermeyer, Hermann Ney, and Ralf Schlter. 2015. From feedforward to recurrent lstm neural networks for language modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3).

Hao Zhao, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory.