Using a Hidden Markov Model, Extended with WordNet, for Textual Based Emotion Detection in a System Initiative Dialog

Hunter Johnson
Colorado School of Mines

INTRODUCTION

In recent years, as human robotic interaction advances certain aspects of the field that seem too abstract and too impossible are becoming a reality. One of those such fields is the recognition of affect based on textual language. The foundation of this sub field of linguistics lies on the fact that words and combinations of words contain more meaning then their dictionary definition. When properly put together words cause an emotional affect on the reader and this affect carries importance when transferring a concept at an emotional level. The future of advanced human robotic interaction will require an integration of a speaker's emotional affect for truly seamless interaction between humans and robots.

In this paper I use a Hidden Markov Model to analyze a corpus of classic fairy tales annotated for affect. Using the trained HMM I then integrate emotion detection into an extremely basic system initiative dialogue management system. Although exceedingly simple, the algorithm used to handle emotion detection into a basic dialogue system could be expanded to larger more meaningful human robotic interaction. Knowing that the jargon used in classic fairy tales does not necessarily apply to today's language Stanford's lexicon WordNet[2] was used for expanding the capabilities of the trained HMM.

The layout of this paper is as follows. In section two I will discuss the motivation behind my work as well as related works. The next section I will discuss my Hidden Markov Model along with its expansion using WordNet [2]. Following this I will be discussing my approach for using this with system initiative dialogues and a rough implementation. Finally, in my discussion and results section, I will describe the pitfalls of my approach as well as further expansions of my work.

MOTIVATION AND RELATED WORK

The problem that will be addressed in this paper contains two parts, the first of which involves extracting and utilizing emotional information from text. The birthplace of affect analysis, or emotional analysis, comes from the sub-field of sentiment analysis. The first notable approach of

sentiment analysis used a polarity data set in which 2,000 movie reviews were assessed for their polarity using Support Vector Machine classifiers [4]. Sentiment analysis requires that a given text undergoes the process of assigning a positive or negative label that captures the text's opinion towards its subject matter. Since Pang and Lee first approached the problem an extension of their work culminates in the Semantic Orientation Calculator in which more than just statistical methods are used such as negation, intensifiers, irrealis blocking, and parts of speech. Many of these corpora fall victim to a data sparsity problem in which training on one corpus does not transfer to other corpora [6]. This issue is particularly bad when the subject matter is expanded from just positive and negative orientation to a full spectrum of archetypal emotions. While the difficulty comes from the requirement of great amounts of human time and effort for tagging some recent works attempt to fuse different data sets. [5] A sort of fusion of sentiment analysis and full affect analysis can be seen in Cecilla Alm's dissertation in which she describes a three-level hierarchy of affect [1].

SOLUTION

Description of Solution

The following section will consist of the following parts. First, the data set used will be discussed in detail. Then the specifics of my Hidden Markov Model implementation will be laid out in a formal manner along with its integration with WordNet, to make it more applicable to out of domain sources. Lastly, an application of emotion recognition in an initiative dialogue management system will be presented.

Dataset

This project begins with the data set used in Alm's dissertation. [1] The data set contains three sets of childrenâĂŹs stories: Beatrix Potter, H. C. Andersen, and the Brothers Grimm. Each of these sets of children's stories was annotated by four readers for their emotional content. Over the course of the annotations six separate people were used. For each of the sentences four annotators choose from the following labels to express the affect of the text: happy, + surprised, - surprised, fearful, sad, angry-disgusted, and neutral. Overall there were 15302 sentences however most of the sentences were labeled as neutral. When reading in the data, words were stripped of punctuation and paired with their parts of speech. Due to the number of neutral sentences dwarfing the number of sentences causing affect, the over all mood of the sentence was labeled as the greatest agreement between the four annotators that was not neutral. In the event of a tie a random affect between the two competitors was chosen. For a sentence to be tagged as neutral it was required that all four annotators labeled the sentence as neutral.

Another important aspect of the data set was the three-level hierarchy of affect labels seen in **Figure** 1. This hierarchy was important because it maps the non-neutral labels for affect onto the

two tags positive and negative so the classification between the labels is less fuzzy. For example, the jargon in sentences labeled with fearful and âĂŞsurprised are more likely to be similar than words in the categories positive and negative. It should be noted that at all levels of the hierarchy include neutral as a tag.

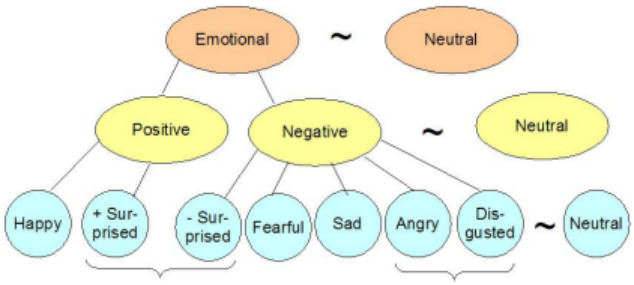

Figure 1: Three-level hierarchy of affect labels.[1]

This corpus was chosen because of the expressiveness of fairy tales, the human exposer to the tales, and the part of speech annotations that came with it. Fairy tales are some of the first stories that humans experience as children and from those stories a person learns and connects certain words to different affects. The theory between choosing this data set as opposed to others lies in the philosophy that to train a model to know the affect that humans feel while reading it the model must learn from the same content as humans do. One last reason that the data set was picked as opposed to others came from the fact that it came pre-tagged with parts of speech from Penn StateâĂŹs Tree bank project. This was important because of the theories presented in Zhu, Li, Chen and Zhou regarding parts of speech.

The HMM

The Hidden Markov Model implemented for this project aimed at tagging the sentences in a sequence word by word where the hidden states are the emotion of speech tags. More formally my Hidden Markov model is defined as follows:

$$E = e_1, \cdots, e_n$$

where $e_i \in E$ and E is the set of all possible hidden states or emotional tags. The transition probability matrix, A, is defined for each $a_{i,j}$ as the probability of moving from state i to state j such that $\sum_{j=1}^{n} = 1 \ \forall i$. In this model that was the probability that given a word was tagged with a specific emotion the next word would transition to any given emotion. A sequence of observations $O = o_1, o_2, \cdots, o_n$ was the sequence of words in sentence. This can be thought of as the input sequence or a sentence broken into each word. The observation likelihoods, also called emissions probabilities, were defined as $B = b_i(o_t)$. Where B is the probability that given a particular observation o the model should be in a particular state e. This piece is commonly represented as a table or a two dimensional array. Lastly, we must define an initial state vector $\hat{q}$ as being the probability that any particular input sequence O will begin in a particular state e. For this model the value q_0 represented the probabilities of a transition to a particular first emotional tag.

Due to the nature of Hidden Markov Models corresponding to sequences of words it was necessary to apply emotional tags to every single word in the data set. This was done in the simplest manor by just applying the dominate emotion of a sentence to each word in the sentence. Then the input sequence was each given sentence broken up into words which were observations. The act of decoding or tagging a sentence with a emotional tag given the model was done using the Viterbi Algorithm which is an exercise in dynamic programming.

In my implementation the initial state transition vector $\hat{q}$ was trained using the following,

$$q_i = \frac{w_{e_i}}{w} \forall e_i$$

Where w_{e_i} is the total number of words tagged with a given emotion e_i in the training set and w is the total number of words in the training set. The transition probability matrix for my dataset was built in the following way,

$$a_{i,j} = \frac{w_{e_i,e_j}}{w_{e_j}} \forall i, j$$

where w_{e_i,e_j} is a transition for a word from emotional state i to emotional state j and w_{e_j} is the total number of state transitions for a given state j. Finally, the observation likelihood table was found as follows,

$$b = \frac{W_{j,e_i}}{w_j}$$

In English that the observation likelihood matrix consists of for every word and every possible tag, the number of times the word appeared with that tag divided by the total number of times the word appeared. It should be noted that this definition for the observation likelihood means that it is a contextually weak, unigram model.

several algorithms exist for performing the decoding stage of solving HMMs but for simplicities sake I chose the Viterbi algorithm which is a a greedy dynamic programming algorithm. In order to tag a sequence observations o_i you first start from the initial hidden state and then calculate the most probable path through the graph while taking into account the input observations. At each step a recursive value is calculated called the Viterbi value which is defined as the following,

$$v_t = \max_{i=1}^{N} v_{t-1} a_{i,j} b_i(o_t)$$

and the initial state is v_0 is given by,

$$v_0 = \max_{i=1}^{n} q_i b_i$$

where i refers to each possible hidden state in the model. In less mathematical terms and applied to my model this definition means that we transverse the graph in which each node corresponds to an emotional tag. On each step we take the current lowest last probability, multiply that by the probability of the next word being in a particular state and by the probability that the given state will transition to the next state. The probability that a particular state will correspond to a given word is trained into the model and given in the observation likelihood table defined above.

The transition probabilities from one state to another is given in the transition probabilities matrix also defined above. Finally for the initial Viterbi value, the probability that the first word will be given a particular tag comes from the initial state probability vector $\hat{q}$ multiplied by the first words likelihood for being in that particular state. On each iteration a pointer is maintained to the state that produced the highest Viterbi value. When all the observations are transversed tagging each observation is a matter of performing a trace back step. This is simply following the pointers from the observation that produced the last greatest viterbi value. The state that the model was in at that point is the emotional tag of that observation or word.

After each word was tagged with a given emotion the sentence was mood averaged in which the most common emotional tag for the sentence was assigned to the entire sentence. When words were not found in the observation likelihood table it was necessary to assign the word an arbitrary value for being in any given state. This meant that an equal value for being in a given state was assigned in the observation likelihood matrix.

Many of the past works in affect, and sentiment analysis note that problems exist when training on one model and then moving to another domain. For this reason WordNet was used for its synonyms in an effort to expand the effectiveness of the trained model across multiple domains. The data sparsity problem manifests itself in two ways. Some words could possibly be seen only once in the entire data set causing their observation likelihood table entry to be less than affective. When observed in an input sequence the word would be required to be emotionally tagged as the only way it had been seen before. This becomes a problem because words can of course have different affects in different contexts. This problem is only solvable through the use of a larger more encompassing data set. The second problem is that some words just do not appear in the data set. In my algorithm a novel approach was taken in an attempt to mitigate this problem. Synonyms are words that are sightly different but when exchanged for each other in a sentence would not change the meaning of the sentence. This was an opportunity to attempt to mitigate the problem of words not appearing in the dataset. The algorithm expanded the observation likelihood table by iterating through all the words in the table and then adding synonyms and simply using the same values for the probabilities of a given word being in a particular hidden state.

My model was deeply flawed in the way that it considered words that clearly do not carry any emotional content. Words such as these are prepositional phrases and other necessary filler words in the English language. In an earlier paper on sentiment analysis the authors described the thought that the only parts of speech that carry emotional force are verbs, adverbs, and nouns. [3] With that in mind the model could be improved. If certain words carry no emotional force why even incorporate them into a model whose purpose is to tag words and thus sentences for emotional affect? This lead to filtering the input data by removing all words that were not elements of the set of all verbs, adverbs, and nouns. This was made easy in the fairy tale corpus because the data came pre-annotated with parts of speech tags. The filter just simply iterated through every word in the data set and removed any words that in theory carried no possible emotional affect.

Due to the problem that more categories means less training data for each category, in a finite set of training data. The model was trained and produced for both the lowest level of the three-level hierarchy of affect and one level up in the hierarchy of affect, which corresponds to negative

labels and positive labels. Experiments were run for both levels of the hierarchy under the assumption that categorizing into three bins would be more accurate that categorizing into 9.

Applying My Algorithm to a Simple Initiative Dialog

In order to show the worth of the emotion detection, a simple model was constructed to enact an extreme version of using emotion detection. Due to time restrictions this devised model is both delicate in nature, extreme, and simplistic. That being said, the underlying principles for the model could be applied to a larger, more robust system in which the emotion detection plays a small but integrated role in the dialogue. This initiative dialogue was treated as a graph in which each node represented a given question in the conversation. For the system in this paper the conversation was applied to a pretend survey center. In this call center each node corresponded to a given question that the call center could ask a person to collect semantic content. In this example the call centers imaginary goal was to collect semantic emotional content on how the speaker feels about certain features of MIT's robot Kismet, seen in **Figure** 2.

Figure 2: Kismet the Robot!

To enact this scenario out the ADE robot architecture was used. Questions were printed to the console and answers are given through speech. The speech is then transformed to text by the sphinx component and sent off to the TLDL parsing component. At this step TLDL parses for semantic content and sends this contend to a new component developed specifically for this given use. The Call Center Application Component is connected to a Dialogue Management Component which given a particular utterance will submit the content to the belief component. The Call Center Component in this application controls a graph made up of nodes. Upon arrival to the TLDL parser before content is parsed a full version of the utterance is sent to the Call Center Application Component. In the component the HMM is used to extract the emotional affect of the statement. Each node must not only correspond to a question but also to some semantic content of which the emotion of the answer must apply. After the emotional context is extracted and applied it is applied to this semantic content and this is submitted directly to the Belief Component. In an effort to make the conversation even more emotionally driven each node contains a set of nodes in which the emotional affect of the answer to the parent nodes question dictates the transversal to the next node in the tree. For this project this framework is lightly exploited and does not consist of an

entire dialogue system, just enough to show a transversal of the tree structure through the use of emotional affect. Lastly, it should be noted that the ultimate goal is to ask questions about Kismet and record their answers for semantic content for latter use.

Experiments and Results

In order to evaluate the effectiveness of the Hidden Markov Model eight separate runs were were used to evaluate different metrics for the different variations of the model. The baseline was given by no data transformation and with tags being on the third tier of the hierarchy of affect labels. The next model was plain sentences but trained and classified on the second level of the affect label hierarchy. These two processes were then repeated for data which was filtered to contain only parts of speech with emotional affect. After this the same four models were trained for models in which synonyms were used to expand the observation likelihood table. The models were trained on 70% of the data and tested for correctness against the remaining percent. Below is **Table** 1 which contains all the necessary information for evaluating the effectiveness for all the different models evaluated.

Overall small amounts of variation occurred when switching between different strategies for manipulating the data and extending the model. The ratio of correctly tagged sentences stayed relatively steady through out all models. This can be attributed to the overwhelming dominance of neutral sentences in the dataset. This dominance caused an extremely high value for the initial state probability vector to be in the neutral state. Because of this fact many of the first states in the observations sequences were neutral. This then compounded because of the nature of how the transition probability table is created as well as how the the training data was tagged. The greatest flaw in this implementation of a Hidden Markov Model was that the tags for individual sentences were expanded to be the tags for each individual word. This caused the initial state of the model to almost always be locked in place because the transition probability table would dominate the Viterbi values. knowing these two facts the Hidden Markov model rarely tagged the hidden state of a word to be anything other than neutral. This realization caused the metric of correctly tagged non neutral sentences seen in the table below.

The greatest number of tagged non neutral sentences occurred for the filtered second tier model with the synonym extended data not making any difference in effectiveness. This most likely occurred because the dominate emotion was taken from the tagged sequence of observations. This meant the second tiered models dominated the third tier models. In the third tier models contention could have occurred between the different sub states. For example a mixed tagged sentence involving neutral, happy and +surprised tags could be tagged as neutral because the happy and +surprised tags are in contention. The same sentence in the second tier model would be tagged as positive because both the happy and +surprised tags would map up to positive tag and then dominate the neutral tags in the sentence. Filtering helped in this effort because when sentences involve words that carry no emotional context that is one more word tagged neutral and thus contributing to the dominance of the neutral tags.

The results in a domain specific training set show that extending the sentences using WordNet had no effect. This makes sense for the way that the data was tested. It is quite likely that data trained on fairy tales will contain jargon that other fairy tales would use. Not to mention that a

writer is likely to use the same words he or she has before when trained on a good portion of their writings. The positive results for extending the model with word net comes from the increase in observation likelihood entries. The size of the table before and after its synonym extension can be seen in **Table:** 2. The table shows that the size Was drastically extended and that should be taken into consideration despite the lack of decrease in the ratio of unknown words between models. Over all including this did not affect performance so the extended filtered model trained on all of the data was used for the emotion recognition in the initiative dialogue system.

Model	Ratio Correct	Sentences Correct not Neutral	Ratio of unknown
Plain Sentences	0.7111	45	0.0762
Plain Sentences Second Tier	0.7062	72	0.0762
Plain Sentences Filtered	0.7138	45	0.1253
Plain Sentences Filtered Second Tier	0.7109	81	0.1253
Extended Sentences	0.711	45	0.0762
Extended Second Tier	0.7062	72	0.0762
Extended Filtered	0.7138	45	0.1253
Extended Filtered Second tier	0.7109	81	0.1253

Table 1

Model	Initial size	Extended Size
Extended Sentences	10627	19806
Extended Filtered	10262	19517

Table 2: Increase in words recognized by the HMM

Their is no formal statistical way to evaluate the initiative dialogue system in this project so a subjective one must be presented. The system functioned about as expected where it traversed the tree properly, stored semantic content, and asked the next questions according to the emotion of the last question. The algorithm however like many initiative dialogue systems fell victim to a few classic problems and hardships. The first of which is the need to have sphinx and TLDL dictionaries for all the possible responses. This problem was not approached in this project because it was not the focus. It cannot however be ignored. The next problem was that the formation of the graph of nodes including the questions and semantic content was incredibly contrived and seemed like an inefficient way to collect and analyze the information. One of the last problems came from the HMM model in which to get the model to tag something as other than neutral required extremely powerful sentences, that were so full of description, that no one would ever talk in that manor. The transitions from nodes using the emotional context of the previous response was also to extreme of an example to ever be useful in a spoken dialogue system. The point of this system, however, was not to make a ready to use system. Something like that would take months if not years. The reason for constructing this was to show that emotional context can be used in a spoken dialogue system. So despite the initiative dialogue system being contrived, and ineffective it was successful in proving that emotional context can be taken into account in a task based dialogue system.

Improvements on Methods

The method for applying an HMM to emotion recognition could be improved in many ways from how it was applied in this paper. The first of which is to apply a bigram or trigram HMM the this corpus which would better encapsulate context. This would have the added bonus of implicitly including such devices as intensifiers and negators. A larger more diverse data set would also help with the problem of the lack of words and situations. The extension of the observation likelihood table could also be applied to a bigram or trigram HMM in the way the the last two or three words, used for context, could permeate using all the synonyms of the last tow or three words. The most crucial improvement, however, would need to come from the handling of the neutral problem. This could possibly be solved by, instead of using a HMM, using a different machine learning model to classify the data.

The system initiative dialogue model could have been improved by using a better emotional classifier as well as the system being less heavily dependent on emotion analysis. It is a mistake to think that the initiative dialogue system here could ever be used in reality. The system is too structured and too reliant on the emotion of the dialogue. In reality a spoken dialogue system should only consider emotional context instead of relying on it to dictate the conversation. Dialogue systems may not necessarily be complex enough yet to consider affect but they are headed there. The necessity of emotional classification into human centered dialogues will be an important part of making a seamless interface for linguistic human robotic interaction.

Conclusion

In this paper a basic dialogue management system was created to collect information on Kismet the robot in which it captured the emotional effect of a persons description of different characteristics of Kismet. The emotional classification of their statements relied on a Hidden Markov model trained on on a corpus of emotionally tagged classic fairy tales. Several methods we attempted to produce a better Hidden Markov Model such as expanding the observation likelihood table with synonyms from WordNet, filtering the texts for only words with the possibility of emotional affect, and reducing the number of classes by ascending the affect label hierarchy.

REFERENCES

[1] Ebba Cecilia Ovesdotter. Alm. *Affect in text and speech*. PhD thesis, 2008.

[2] Mark Alan Finlayson. Libraries for accessing the princeton wordnet: Comparison and evaluation.

[3] Dan Jurafsky and James H. Martin. *Speech and language processing*. Pearson Education, 2014.

[4] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and TrendsÂő in Information Retrieval*, 2(1âĂŞ2):1âĂŞ135, 2008.

[5] Zhu Suyang, li Shoushan, Chen Ying, and Zhou Guodong. Corpus fusion for emotion classification. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, page 2387âĂŞ2397, Dec 2016.

[6] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267âĂŞ307, 2011.