Affect - Language Model

Quentin Bacuet
quentin.bacuet@epfl.ch

***Abstract*—In this project we trained a special type of Language model, the Affect-Language Model that take into account the affect part of the sentences. This model will be used to perform sentence generation for a specific affect category (Anger, Positive Emotion,...). The model was trained on two different dataset with two different architecture using LSTMs. Finally, human testing was used to assert the validity of both of them.**

I. Introduction

A language model is a probability distribution over sequences of words, it has many use case such as speech recognition or machine translation for instance. A lot of the past studies mostly concentrate on sentence structure, word representation or semantic without considering much the affect part of the sentence. The affect is a term that take in account the emotion, the mood and the feeling [1]. We will use the affect part of the sentence to create a new type of language model called Affect-Language model (Affect-LM). Classic language model can be used to generate sentence, this process can be view as a function mapping a sentence to a word (the word that is more probable to complete the sentence). With the affect-LM we propose a new function that will take as input a sentence but also an affect part a_{t-1} and a parameter β that will control the affect strength in the generated word (how much affect we want in the word) and output a word. This could improve all types of language models that could later be used for more general purpose such as bots. Hence we propose in this paper to answer to two different questions:

1) Can we generate sentences with specific affect categories and specific strength β using the Affect-LM?
2) Does adding the affect categories as an input to the language model actually improve the grammatical correctness, structure and meaning of the generated sentences? What about the perplexity score? In other word is the Affect-LM superior to the more classic LM?

To do this we will first present the datasets that we will use in section II, we will then explore the datasets to get some basic statistics of them in section III, we will then define the metrics that we will use in section IV, then we will define the models and architecture that we will implement in section V, then we will describe the pipeline that we used in section VI and finally in the last section VII we will present our results.

In this paper, we based ourselves on "Affect-LM: A Neural Language Model for Customizable Affective Text Generation" [2] but we tried to add more details on the procedure, pipeline and the various models used.

II. Dataset Presentation

For this project we used a collection of various texts. The first dataset used is the Cornell Movie Dialog Corpus which is available for research purpose [3]. It is composed of dialogues lines of fictional conversations extracted directly from movie scripts.
Additionally, a second dataset was used, the reviews of 1,000 different hotels. This dataset is available on Kaggle [4]. It comprised more than 35'000 reviews of real user of their stay at the hotel.
Finally, for the affect part, we used the Linguistic Inquiry and Word Count (LIWC) dictionary [5]. This program allowed us to map each words in the corpus to various categories such as the tone, its grammatical type and its affect category. The latter one is what we will use throughout this report. Threw this paper, we used the 5 following affect categories: positive emotion, angry emotion, sad emotion, anxious emotion and negative emotion. We can see in table I some examples, we can note than some words have multiple categories.

Word	Pos. Emotion	Neg. Emotion	Sad	Anxious	Anger
threat	0	0	0	1	1
misery	0	0	1	1	0
love	1	0	0	0	0
hate	0	1	0	0	1

TABLE I: Example of output from the LIWC for different words

III. Exploratory analysis

In this section, we will describe the fields that we will use as well as some basic statistics for both datasets. For the cornell dataset, we will use the file $movie\_lines.txt$. This file contains the actual text of each utterance. For the hotels reviews, the reviews are directly available at the field *reviews.text*. We will now present some basic statistics of the two text datasets.

A. Cornel Dataset

The cornell dataset is comprised of 304,713 different sentences and 62,795 different words when we remove the punctuation and we don't take into account the cast of the words.

We can see in figure 1 that the large majority of the number of words per utterance is smaller than 40. Indeed the mean of this distribution is 10.84 words with a standard deviation of 12.97 and a median of 7 words.

We then plotted a wordcloud of the most common words and bigrams in figure 2.

Finally, we will describe the emotion composition (in approximation) of the Cornell dataset in table II below.

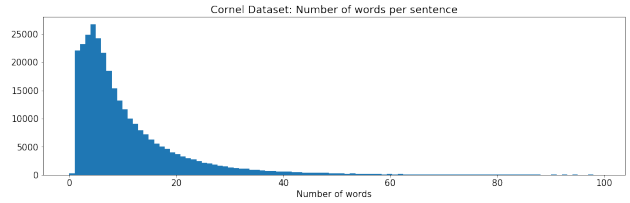


Fig. 1: The number of words per sentence for the Cornell dataset. For visibility, we limited the x-axis to 100 words.

((a)) Unigram ((b)) Bigram

Fig. 2: Wordcloud of the most frequent words and bigrams for the Cornell Dataset

Emotion	Percentage of words
Pos. emotion	3.2 %
Sad	0.4 %
Anxious	0.2 %
Neg. emotion	2.3 %
Anger	1.0 %
Colored Words	5.5 %

TABLE II: Percentage of words with emotion in the Cornell Dataset

B. Hotel Reviews Dataset

The Hotel dataset is comprised of 118,933 different sentences and 36,958 different words when we remove the punctuation and we don't take into account the cast of the words, each of those reviews also contain a grade from 1 to 5. From this dataset, we only kept the angry reviews. Hence we filtered out all the reviews for which the grade was bigger or equal to 3. After this, we have 24,587 different sentences with 15,046 unique words in total.

We can see in figure 3 that the large majority of the number of words per utterance is smaller than 40. Indeed the mean of this distribution is 12.73 words with a standard deviation of 6.34 and a median of 11 words.

We then plotted a wordcloud of the most common words and bigrams in figure 4.

Finally, we will describe the emotion composition of the Hotel dataset in table III below. We can see that compared to the Cornell dataset, the hotel dataset is more colored and contains more angry and negative emotion. This was expected as we only kept the reviews with very bad grades and hence angry customers.

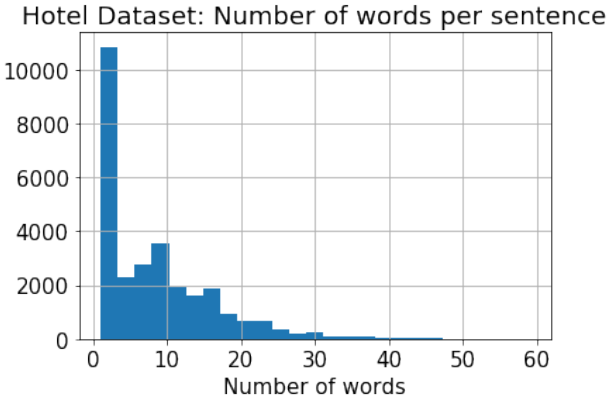
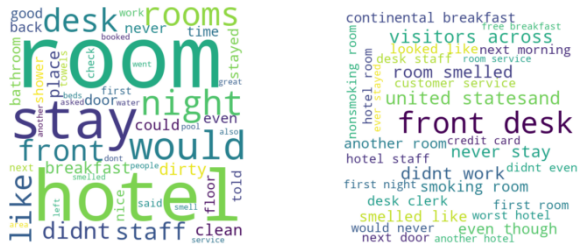


Fig. 3: The number of words per sentence for the Hotel dataset. For visibility, we limited the x-axis to 60 words.



((a)) Unigram ((b)) Bigram

Fig. 4: Wordcloud of the most frequent words and bigrams for the Cornell Dataset

Emotion	Percentage of words
Pos. emotion	2.2 %
Sad	0.5 %
Anxious	0.4 %
Neg. emotion	3.2 %
Anger	1.2 %
Colored Words	6.1 %

TABLE III: Percentage of words with emotion in the Hotel Dataset

C. LIWC words

In this subsection, we will explore the words that we kept for the analysis (we describe the process of the filtering of the words below). As we can see in the table IV below, there is a lot more words with negative emotion than for the other of emotions.

Emotion	Percentage of words
Pos. emotion	4.0 %
Sad	0.8 %
Anxious	0.9 %
Neg. emotion	5.1 %
Anger	1.9 %
Colored Words	9.0 %

TABLE IV: Percentage of words with emotion in the words used

IV. Metrics for Performance Evaluation

The evaluation of a machine learning model is one of the most important part of any project. The choice of metrics is crucial and will influence how the performance is measured and how we can compare between different models. Throughout this report we will use three different metrics.

A. Perplexity Score

The first metric that we used is the perplexity score. It is a standard way of measuring the empirical performance of a certain model on a specific dataset that can be directly used to compare between various models. More formally, it is a measurement of how well a probability distribution predicts a sample. It is defined as:

$$P(p) = 2^{H(p)} = 2^{-\sum_x p(x)\log_2 p(x)} = 2^{cross\ entropy\ loss}$$

It is also two to the power of the cross entropy loss. Hence the smaller the perplexity score is the better it is. Indeed if a model A has a smaller perplexity score than a model B on the same dataset, this indicates that the model A predicts the dataset better than the model B.

B. Accuracy

The second one is the accuracy of the model. It measures the proportion of cases in which the model agrees with the ground truth and it is also used as a standard metric to rank the different models. In our case, as we will predict the next word out of thousands of unique words, accuracy can typically be smaller on our model than for instance on a classic binary classification.

C. Human Test

Finally the third metric is the human test. This we will be used to verify if our model can generate correctly sentences with respect to the grammar and to the emotion chosen (we described the sentence generation process below). For this we evaluate 100 sentences. The first 33 are generated from the Affect-LM model trained on the Cornell dataset, the next 33 from the baseline model trained on the Cornell dataset and finally the last 34 from the Affect-LM on the hotel dataset. Those sentences are picked randomly from sentences generated available in the appendix H. We then randomized the order of the sentences and added them in a Google Form. We then asked 3 different paid judges 6 question per sentence:

- Is the sentence grammatically correct?
- Does the sentence contain positive emotion?
- Does the sentence contain negative emotion?
- Does the sentence contain angry emotion?
- Does the sentence contain sad emotion?
- Does the sentence contain anxious emotion?

For each of those questions, the judges answered using a five-level Likert scale: Strongly disagree (1), Disagree (2), Neither agree nor disagree(3), Agree(4), Strongly agree(5). This is a really important metric as it can be directly used to give real insight on the model performances. As in the end our model is used for sentence generation and that only human can be real judge of their quality.

D. Validating Models

When it comes to validating models, we usually perform k-fold cross-validation. However, in the case of neural networks and LSTMs, training can easily take hours/days, even on powerful configurations. This is why we opt instead for a train/validation/test split of the data. At the beginning of the training process, we randomly partition the initial training set into three separate sets containing respectively $75\%, 15\%$ and 10% of the data. We will use this for all the dataset and models.

V. Models

We now introduce the models used for the text generation task and the creation of a language model. It is important when comparing the performance of models to consider a baseline performance which will serve as a benchmark for the other models. We fix ground zero to be the perplexity score of the baseline model V-B. In I, we asked ourselves if adding the affect part to the neural network make the overall model superior to common models on text generation tasks. One guess why adding the affect part could improve the model is that it could be more able to capture the underlying structure of the sentence than the classic LM as sentences are likely to be coherent in terms of emotion. Indeed, if a sentence starts with positive emotion word, it will most likely have an ending that will also contain positive words. On the other hand, if a sentence starts with negative emotion word it will most likely have an ending that will also contain negative words. Hence the baseline model would have the choice between all the words to finish the sentence. But the Affect-LM model V-C could have a range of possible words much smaller as words that are not in the same emotion category as the beginning will be "discarded" (their probability of appearance will be drastically reduced). This explanation is just a guess, we will check in the following sections VII if first of all the Affect-LM model performs better in terms of the metrics defined in IV. To verify this affirmation we trained a baseline model and an Affect-LM model.

All our models take in input a batch of sentences coded using the one-hot encoding and output another batch of sentences in one-hot encoding. The one-hot encoding map a word to a vector with a dimension equal to the number of categories (in our case unique words). This vector have 0 everywhere except at one position where it is a 1. This position is arbitrary and unique for each word. The main advantage of the one-hot encoding compared to other encoding is that the model will not assume a natural ordering between categories. Hence the input and output dimension of our model are Batch size x Maximum length of the sentences x Number of words.

The models were implemented using the Keras implementation in Tensorflow [6].

A. Layers

In this subsection we will describe all the layers that we will use in our architecture.

- Input Layer: This layer is simply a layer that will be used as a placeholder for the input sentences.
- Masking Layer: this layer will be used as sentence don't have necessarily their length equal to the maximum length of the sentences. This is used to make sure that the loss function will only count actual words in the sentence and not the zeros added by the zero padding (see more details in VI-B).
- Embedding dense layer: this layer will be exclusively used to reduce the dimension in the network. Indeed, as explained above our input consist of one-hot encoded vectors that are in really high dimension. Hence if we want to ease and accelerate all the learning process, we must reduce it by using an embedding layer. This layer is a simple matrix multiplication with no activation function with an output size smaller than its input.
- LSTM: this layer is an extension of the classic RNN that doesn't suffer from the problem of the vanishing gradient [7].
- Sigmoid time distributed layer: this layer is a classic sigmoid layer using weight sharing.
- Concatenate layer: this layer is very straightforward; given two input it will simply concatenate them. The output dimension is then the sum of the two dimension of the inputs.
- Output time distributed layer: this layer is a classic softmax layer using weight sharing.

We can note that as our data is 3-dimensional, every layer (except the LSTM layer) use weight sharing; the same layer and weights are applied to every "word" in the sentence.

B. Baseline

The baseline model is a classic model for text generation using LSTMs. The architecture is composed of 6 different layers illustrated in the figure 5 below.

1) An input layer of size (batch size, maximum length of the sentences, number of words).
2) A masking layer of size (batch size, maximum length of the sentences, number of words).
3) An embedding dense layer of size (batch size, maximum length of the sentences, 200).
4) A first LSTMs layer of size (batch size, maximum length of the sentences, 200).
5) A second LSTMs layer of size (batch size, maximum length of the sentences, 200).
6) An output layer time distributed of size (batch size, maximum length of the sentence, number of words).

Hence the output layer give the following formula with $k_{t-1} = (w_{t-1}, w_{t-2}, ..., w_2, w_1)$ the past words, f_{lstm} the output of the LSTM layer and W the weights and b the bias term the softmax layer:

$$P(w_t = i|a_{t-1}) = \frac{exp(W_i \cdot f_{lstm}(k_{t-1}) + b_i)}{\sum_i exp(W_i \cdot f_{lstm}(k_{t-1}) + b_i)} \quad (1)$$

From this we can see that the matrix W give an embedding (in the sense of Word2Vec) of the words in a lower dimension.

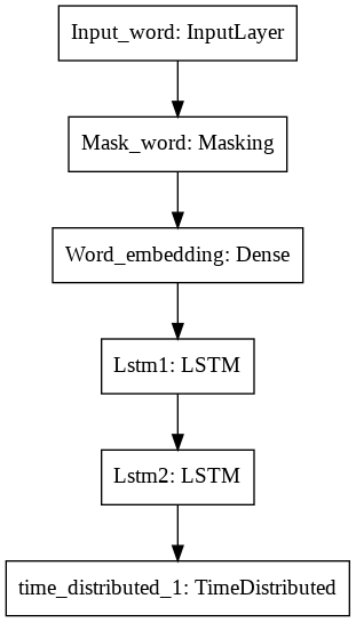


Fig. 5: The architecture of the baseline model

C. Affect-LM model

The Affect-LM model is an extension of the baseline model. The architecture is composed of 2 different model, the baseline model (without its output layer) explained above and a model used for the affect (this is described as "the affect part of the model"). The affect part of the model is described below:

1) An input layer of size (batch size, maximum length of the sentences, number of different emotions).
2) A masking layer of size (batch size, maximum length of the sentences, number of different emotions).
3) A sigmoid time distributed layer size (batch size, maximum length of the sentences, 100).
4) An output layer time distributed of size batch size, maximum length of the sentences, 200).

The output of the affect part is then multiplied with the strenght parameter β. Finally, the output of the baseline and the multiplied output are then concatenated together (both output are of size 200, hence the output of the concatenate layer is 400) and passed as an input to a softmax time distributed layer that have an output size of (batch size, maximum length of the sentence, number of words). The all architecture is illustrated in the figure 6 below.

Hence the output layer give the following formula with $k_{t-1} = (w_{t-1}, w_{t-2}, ..., w_2, w_1)$, $a_{t-1} = (e_{t-1}, e_{t-2}, ..., e_2, e_1)$ the affect categories of the sentence, f_{lstm} the output of the LSTM layer, f_{affect} the output of the affect part of the architecture and W,A the weights and b the bias term the softmax layer:

$$P(w_t = i|k_{t-1}, a_{t-1}) = \frac{exp(W_i \cdot f_{lstm}(k_{t-1}) + A_i \cdot f_{affect}(a_{t-1}) \cdot \beta + b_i)}{\sum_i exp(W_i \cdot f_{lstm}(k_{t-1}) + A_i \cdot f_{affect}(a_{t-1}) \cdot \beta + b_i)} \quad (2)$$

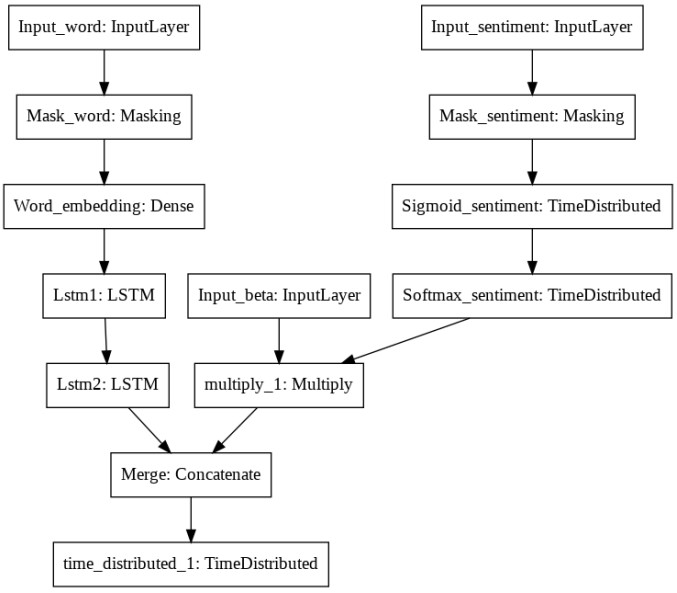


Fig. 6: The architecture of the Affect-LM model

From this we can see that as before in 1 the matrix W give an embedding, but now we also have the matrix A that give an embedding of the words with respect to their affect categories.

Finally, we can see that both models are dependant in term of their number of parameter to the number of words. A large number of words can increase the number of parameters and drastically slow the learning process, hence this number should be relatively small. We can have the same reasoning for the maximum length of the sentence with the number of internal states and parameters for the LSTMs. A large number for the maximum length of the sentences could also the slow the learning process. We summarized all the parameters used in table VI.

VI. Pipeline Description

In this section we will describe step by step the pipeline used in the project, from the creation of the dataset to the sentence generation.

A. Filtering

The first part of the pipeline is the filtering of the words and the sentences from the Cornell and Hotel Dataset. It is crucial to first normalize the sentences. We do this by first removing the punctuation (i.e. '.', ',', '/', '-', ect). We also made sure to remove *'s* at the end of the words to ensure that for instance *boy* and *boy's* are considered to be the same word. Finally, we putted every letter in each sentences to lower case. At this stage we are facing an issue, there is more than 60,000 unique words in the Cornell dataset and we can't run Machine Learning Algorithms with that much words as we have seen in the end of the section V. Hence we want to reduce the number of words in a way or the other without reducing too much the number of sentences. But we can't simply remove words in a sentence as some sentence can rely totally on this word to make sense. Hence we have to remove some sentences from the corpus. We first filter scarce sentences. We defined the scarcity of a sentence as the sum of the number of occurrences of each word in the sentence divided by the number of word in it. Hence it's the mean number of occurrences of each word in the sentence. More formally:

$$S(s, C) = \sum_{word \in s} \frac{O(word, C)}{|s|}$$

with S(s,C) the scarcity function taking as variable a sentence s and a corpus of sentences C, O(word,C) the occurrence function that return the fraction of time the word appears in the corpus C and $|s|$ the length of the sentence. Then we simply take the 95% sentences that are the less scarce. Finally, we remove sentences that are too small (1 word) because they are not really interesting for our case.

B. Data Preparation

In this step we will take the filtered corpus of sentences to an actually trainable dataset that our model can understand (a matrix with number). The first step is to set the maximum length of the sentences. This length shouldn't be too big as explained in the end of the section V. Hence we have to drop sentences that are too big. We define the maximum length of the sentences as the average number of words plus two times the standard deviation of the number of words in the filtered corpus. This ensure that we will roughly keep 95% of the dataset. In our case the $std = 12.03$ and $E = 10.98$, hence the maximum length is 35. Even with the filtering there is still too much unique words, so we only keep the 15,000 most common ones and replace the other one by an $Unknown$ tag.

To continue, at the end of each sentence we add an End tag that will be used by the model to determine when a sentence ends. We can note that those tags count as unique words. After encoding the sentences in one-hot encoding we know have to make sure that each sentence (that is now a matrix of size: size of the sentence x number of words) have the same size to be able to input it to the model in a easy way. To achieve this we use zero padding, this mean that we append vectors full of zero (with a size of the number of words) at the end of the matrix until it has a size of: maximum length of the sentences x number of words; in our case 35 x 15,002.

Now we prepare the affect part obtained from the LIWC dictionary. We generate a matrix for each sentence of size: length of the sentence x number of different affect categories. We create it in a cumulative way. When an emotion is seen earlier in the sentence we will have this emotion for all the words after. More concretely, if we have the sentence: "I like this project" and if the only colored word in it is "like" then the matrix obtained will be as in table .

We can see that all the words after "like" will have the positive emotion, even though they are not colored. The main reason to do this is to ensure emotion coherence in the sentence. This matrix after adding the zero padding is

Sentence	Pos. Emotion	Neg. Emotion	Sad	Anxious	Anger
I	0	0	0	0	0
like	1	0	0	0	0
this	1	0	0	0	0
project	1	0	0	0	0

TABLE V: Example of sentiment matrix obtained from the sentence "I like this project".

of size of maximum length of the sentences x number of different emotion, in our case 35 x 5. It will be used as an input for the affect part of the model described in V-C.

C. Training on the Cornell Dataset

We then trained the affect-LM architecture and the baseline architecture using the Cornell dataset. We split it in 3 subset the training/validation/test set with respectively 75%,15% and 10% of the data. We trained them using the Adam Optimizer for 50 epochs on the train data using a batch size of 64. We did this 3 times with three different beta for $\beta \in [1, 2, 3]$ and picked the model with the smallest perplexity score on the validation set. The training is done by taking as input the sentence without the last word (it will always be the End tag) and as output the sentence without the first word (with the End tag as the last word).

D. Training on the Hotel Dataset

The best model obtained above for both the affect-LM architecture and the baseline architecture are fine tuned for the Hotel dataset. Indeed they are used as seed for the new dataset. The training process is the same as the Cornell dataset.

E. Sentence Generation

For the sentence generation, we used 3 different models. The models obtained from the Affect-LM architecture on both the Cornell and the Hotel datasets and the model obtained from the Baseline architecture on the Cornell dataset.

The generation method that we use will sequentially add a new word at the end of the sentence and feed the new sentence back to the model until a End tag appears, then we stop. We just need to define two thing: how do we pick the affect input for the Affect-LM architecture and how do we actually choose the word from the output layer? For the first question there is two answers, we can either infer the affect part of the sentence by simply looking in the LIWC dictionary or "force" a specific affect category. We will use the latter one, it is achieved by simply putting 1's in the category of desire in the sentiment matrix. We can note that this works better for neutral sentence beginning. For the second question there is also two answers. The output of both architecture is a softmax layer. It will give us a probability distribution over all the words possible (including the two tags). We first remove the $Unkown$ tag probability and normalize the distribution. We then have two choices either take the argmax as the next word or pick randomly the next word using the probability distribution. We generated sentences with both methods. We can note that to mitigate the effect of randomness of the second method we used the method of temperature sampling. It is a method of sampling that take an extra parameter, the temperature τ and in function of it, change the distribution. If τ is equal to 1 then the distribution doesn't change. If τ get closer to 0 then the distribution will move towards the case of the argmax (first method described above), where all the probabilities are 0 except at the argmax where it is 1. Finally if the temperature is bigger than 1, the distribution become smoother. We typically used temperature smaller than 1 to get some randomness without changing the dominating probabilities.

Number of words	15,002
Maximum length of the sentences	35
Number of emotions	5
Batch size	64

TABLE VI: Summary of the parameters used

VII. Results

In this section, we are going to analyze the different results that we got, and based on those, answer our questions in I.

A. Learning curves

We are now going to analyze the learning process of our models and detail the hyperparameter chosen for both the Cornell and the Hotel dataset. As we described before we will choose the β based on the model that perform the best on the validation set.

For the Cornell dataset, we can see on figure (A.a)) that the best hyperparameter based on the perplexity score is actually the model with a $\beta = 3$. We have the same conclusion on the accuracy (see figure (A.c))).

For the Hotel dataset, we can see on figure (B.a)) that the best hyperparameter based on the perplexity score is actually the model with a $\beta = 3$ (it is very close from $\beta = 2$). We have the same conclusion on the accuracy (see figure (B.c)).

For completeness sake, we added in the appendix (section C and D) the perplexity scores and accuracies per model on the validation and the training set.

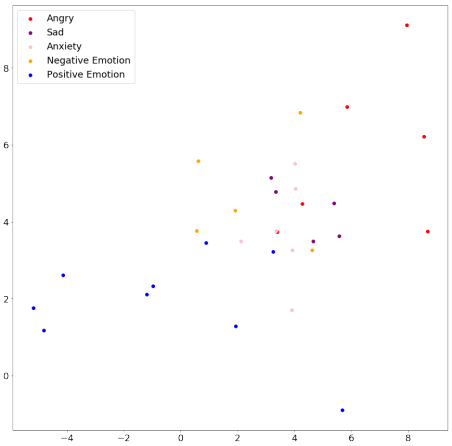
To conclude which model is better and by how much we should use the metrics based on the test set. Hence we see in table VII that indeed the Affect-LM performs better as the perplexity score on both the Cornell and the Hotel dataset is lower for the Affect-LM than for the baseline. This answer a part of the question 2 in I, adding the affect part to the classic language model improve the overall perplexity score of the model. We can note that the perplexity score are low but this can be explained by the number of epochs.

Perplexity score	Affect-LM $\beta = 3$	Baseline
Cornell	13.579	14.636
Hotels	3.531	3.837

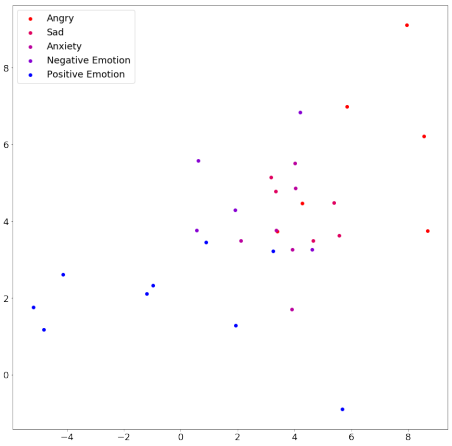
TABLE VII: Metrics on the test set for the Cornell and the Hotel dataset after 50 epochs

B. Sentiment Embedding

In Equation 2, Affect-LM learns a weight matrix A which captures the correlation between the word w_t, and the affect category a_t, and hence will give us an embedding in terms of the affect categories. Thus, each row of the matrix A_i is an embedding with respect to the emotion of the i-th word in the vocabulary. In Figure 7, we present a visualization of these embedding (from the Cornell dataset), where each data point is a separate word, for the sake of visualization we simply visualized a few words from the LIWC dictionary. We can still see that there is indeed clear clusters (the angry and the Positive emotion are clearly grouped together). Even more interesting, we can see from the second plot that there is a polarization of the words from the positive ones to the angry ones. This can also be used to show that we successfully learned the affect part of the sentences.



((a)) Embedding



((b)) Embedding with a diverging colormap

Fig. 7

C. Sentence Generation

We generated multiple sentences with multiple beginnings, they are all in appendix H. For the Cornell dataset, we used 3 different beginnings: "I feel like", "He was so" and "I looked so". For the Affect-LM architecture we generated sentences with 6 different betas and all the different affect categories. Finally, for the Hotels dataset we use 2 different beginnings: "The hotel was" and "The place was". We can see that we actually get good results for most of the sentences and that changing the affect category and the β change the content of the sentence. To be sure of that we will now analyze the results from the Human test.

D. Human Test Metric

We will first describe the agreement between the 3 judges using the Krippendorff's alpha. The alpha for the affect categories was 0.661 and the one for the grammatical correctness was 0.595. This is sufficient to make conclusion from the judge results. We will start by analyzing the grammatical correctness evolution with respect to the β. We will do it by looking at the results of the question: "Is the sentence grammatically correct?". First, we start with the figure 8. It only take in account the results from the sentences generated from the Cornell dataset. The dotted lines (representing each judges) and the red curve are obtained from the 33 sentences generated from the Affect-LM architecture. The score goes from 1 (Strongly Disagree) to 5 (Strongly Agree). The pink and brown curve are obtained from the 33 sentences generated from the baseline architecture. The dotted lines (representing each judges) where then calculated by taking the average of each results for a given β. The mean score is just the mean of the answers of the judges for each of the sentence. The pink and brown curve are respectively the mean score and the median score of the grammatical correctness of the sentences generated by the baseline model. We can see that when $\beta = 0$, the mean score of the Baseline and of the Affect-LM are almost the same (close to 4), this actually show that when $\beta = 0$ the Affect-LM and the baseline model are equivalent with respect to grammatical correctness. When we increase the β however we actually get better results as we can see on the figure. But if it become too big we actually loose grammatical correctness. This could be due to the fact that as we increase the β the model try to push too much affect in the sentence without taking in account the Language Model and hence degrading the overall grammatical structure. But the main conclusion that we have here is that for specific betas, we actually perform better in terms of grammar with the Affect-LM than with the Baseline. We can conclude the same with the Hotel dataset on figure 9. Hence, this answer a part of the question 2 in I, adding the affect part to the classic language model improve the overall grammatical correctness of the sentence generated for certain betas.

We will now analyze the results obtained on the 5 other questions related to the affect category in the sentence. First, we start with the figure 10. The dotted lines (representing each judges) and the red curve are obtained from the 33 sentences generated from the Affect-LM architecture and the Cornell dataset. The score goes from 1 (Strongly Disagree) to 5 (Strongly Agree). The pink and brown curve are obtained from the 33 sentences generated from the baseline architecture and the Cornell dataset. The dotted lines (representing

each judges) are calculated by taking the average of each results for a given β of the question with the same affect as the one that we used to generate the sentence. For example, if we generated the sentence "I like this project" with the affect "Positive Emotion", we will count in the figure only the answer to the question "Does the sentence contain positive emotion?". On the other hand for the pink and brown lines we count every question generated by the Baseline model. We can see that as for the grammatical correctness, when $\beta = 0$ the Affect-LM and the baseline model get affect score that are close together (that are close to 1, i.e. no affect in the sentence). When we increase β we see that the affect score increase meaning that the judges detected on average more the affect that was expected in the sentences. We also did a linear regression and we saw that the mean score of the judges can be approximated by a line with a slope of 0.38; indicating a positive linear relation between the mean score and the beta. We can conclude the same with the Hotel dataset on figure 11.

Finally to make sure that we don't get the same results with the sentences generated by the baseline model we plotted the histogram of the mean score of the answers of the questions related to the affect categories in figure 12 and we can observe that indeed the results are close to 1, indicating that the baseline do not generate most of the time sentences with affect compared to the Affect-LM. We also plotted the histogram of the the answers of the question with the same affect as the one that we used to generate the sentence from the Affect-LM in figure 13. We see in this case that the results are much closer to 5, indicating compared to the baseline that we managed to generated colored sentences. We have the same conclusion with the Hotel dataset on figure 14.

This answer the question 1 in I, we can generate sentences with a specific affect category and increasing the β will increase in average the affect strength of the sentence.

VIII. Conclusion

In this project we have successfully implemented an Affect-Language Model that take as input a sentence but also the affect categories of the word in the sentence and a strength parameter β. Using the human testing we have shown that the model can generate colored sentences that vary in emotion with respect to β. To continue, we also have shown that for specific betas, we perform better in terms of grammatical correctness with the Affect-LM. Finally, we also have shown that we get smaller perplexity score for the Affect-LM than for the baseline model. Hence we can conclude that the Affect-LM is better than the classic LM and that affect part in the sentences should be always considered for LM.

IX. Discussion

In this section we will describe the main difficulties that we encountered throughout the project. The first one is the difficulty of reducing the number of words without reducing too much the number of sentences and filtering sentences that will be bad for the training of the model. This difficulty is classic for Machine Learning, it is a trade-off between the quality of the dataset and its size. The second difficulty was mainly the computational resources and the time. Indeed without a graphic card, training such model in a reasonable time (24 hours) is mostly impossible. Hence we had to use alternative methods such as Google Colaboratory or the IC Cluster. The main advantage of Google Colab is that the service is free and works as a jupyter notebook; hence it make all the coding part really easier. On the other hand, the IC cluster run scripts and not notebooks (hence it can be harder to code without bugs) but it offer on average a faster training time. Finally, the last difficulty was the human testing. It's a really crucial part as it really determine the overall performance of the Language model. But it can be hard to find the good questions in the form, questions that would be easily understood by the judge but that would also make sense for evaluation purpose.

X. Acknowledgment

I would like to express my very great appreciation to Dr. Pearl Pu and Mr. Yubo Xie for their valuable and constructive suggestions during the planning and development of this all research work.

References

[1] Klaus R Scherer, Tanja Bänziger, and Etienne Roesch. 2010. A Blueprint for Affective Computing: A sourcebook and manual. Oxford University Press.

[2] Sayan Ghosh, Mathieu Chollet, Eugene Laksana, Louis-Philippe Morency and Stefan Scherer. Affect-LM: A Neural Language Model for Customizable Affective Text Generation. Institute for Creative Technologies, University of Southern California, CA, USA and Language Technologies Institute, Carnegie Mellon University, PA, USA. 2017

[3] Cristian Danescu-Niculescu-Mizil and Lillian Lee, Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs, Cornell University, 2011. *https : //www.cs.cornell.edu/ cristian/Cornell*$_{M}$*ovie − Dialogs*$_{C}$*orpus.html*

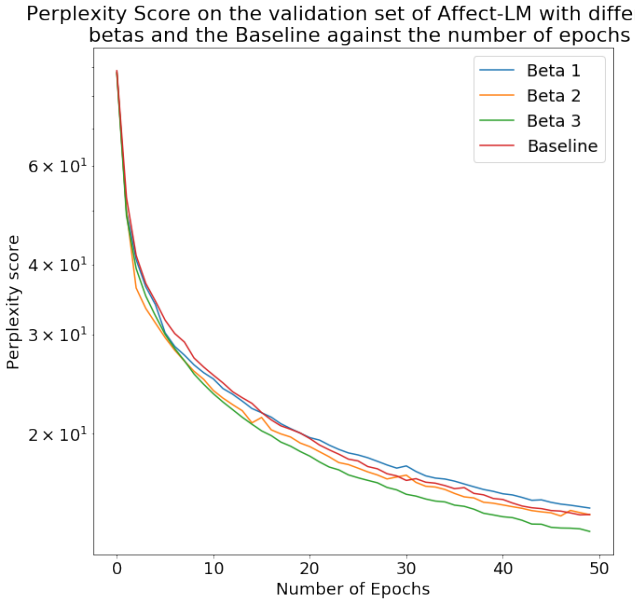
[4] Hotel Reviews, A list of 1,000 hotels and their online reviews, *https://www.kaggle.com/datafiniti/hotel-reviews*

[5] James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: Liwc 2001. Mahway: Lawrence Erlbaum Associates 71(2001):2001.

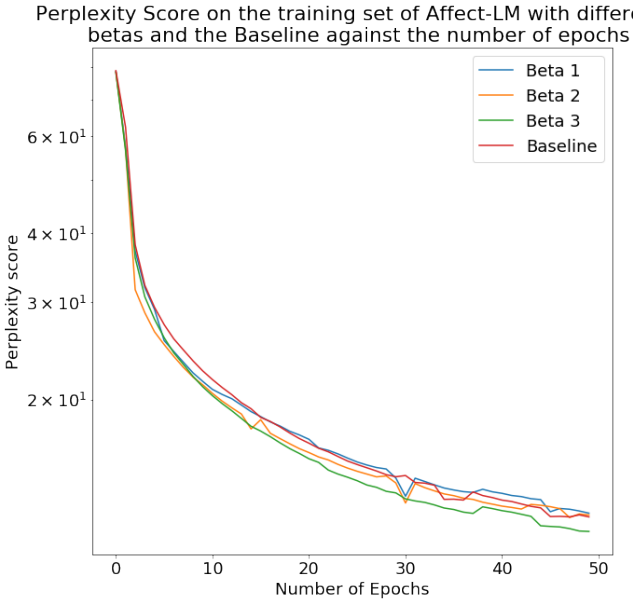
[6] Martín Abadi et al. 2016. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI). Savannah, Georgia, USA.

[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural computation 9(8):1735–1780.

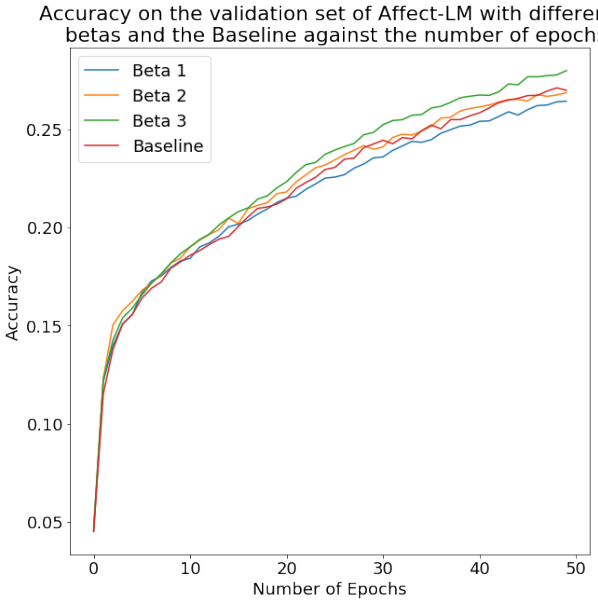
APPENDIX A
LEARNING CURVE: CORNELL DATASET



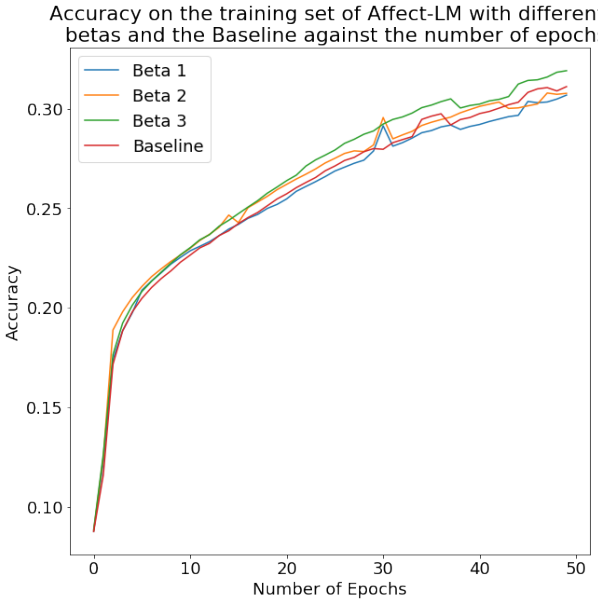
((a))



((b))

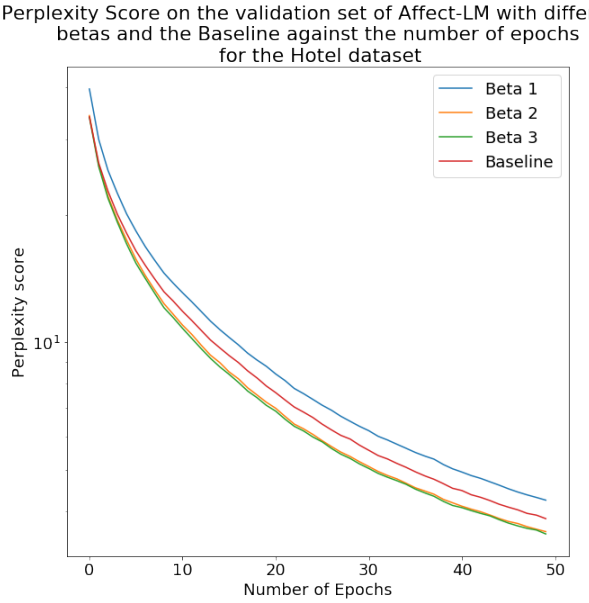


((c))

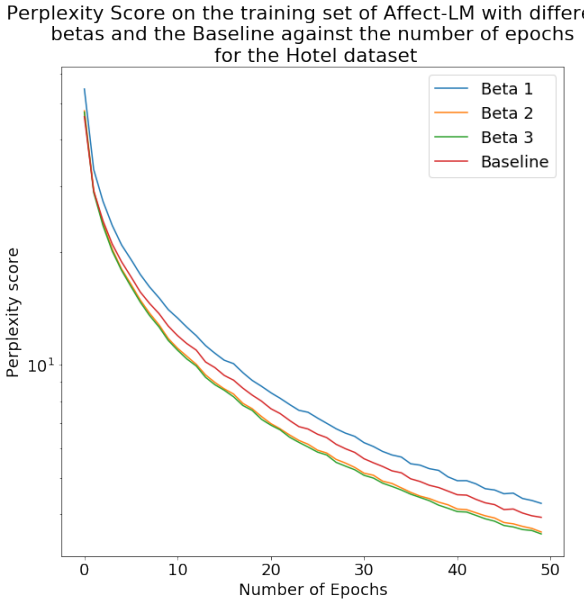


((d))

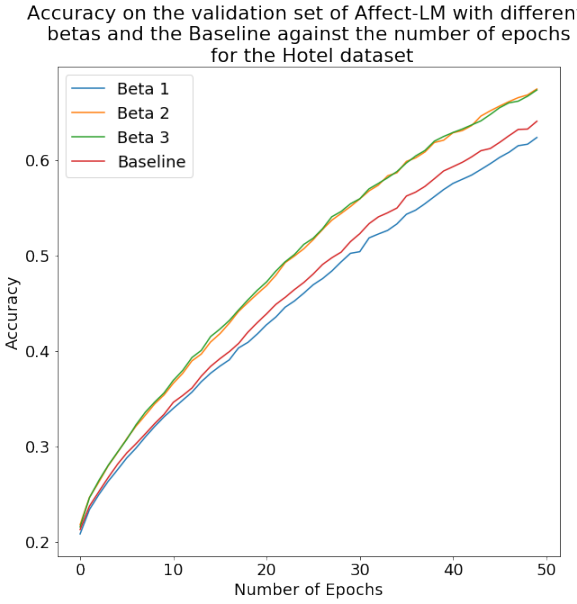
APPENDIX B
LEARNING CURVE: HOTEL DATASET



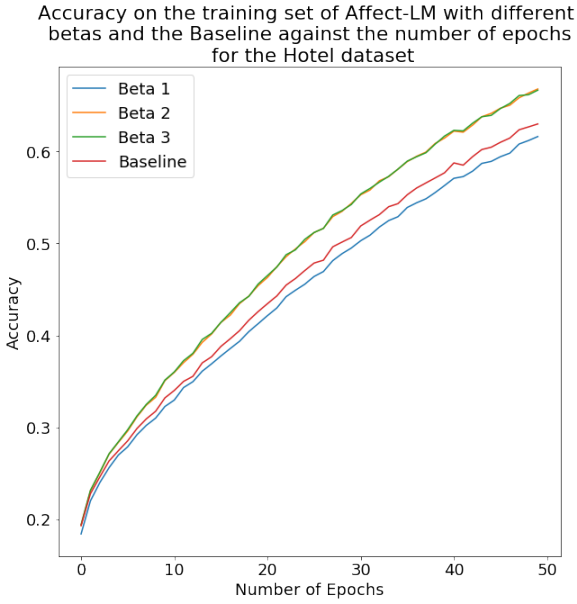
((a))



((b))



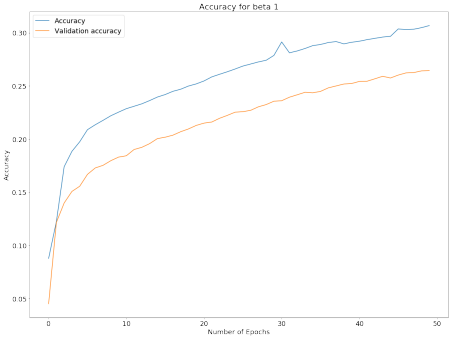
((c))



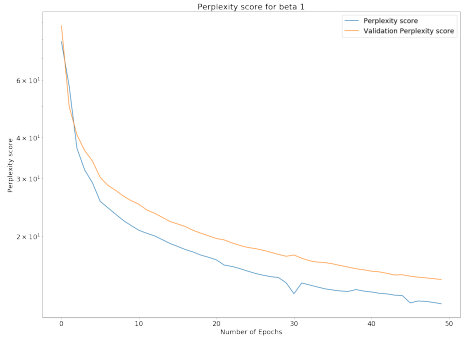
((d))

APPENDIX C
LEARNING CURVE IN MORE DETAILS: CORNELL DATASET

$\beta = 1$:

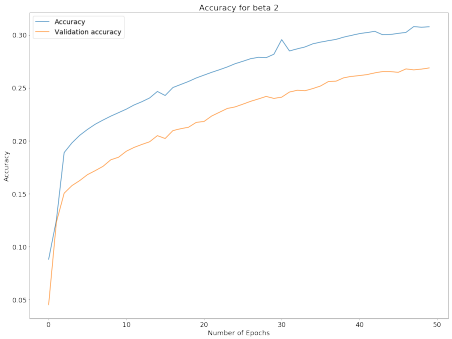


((a))

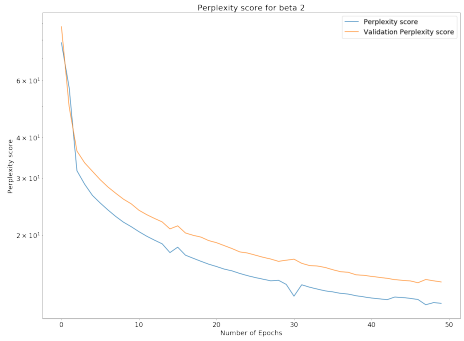


((b))

$\beta = 2$:

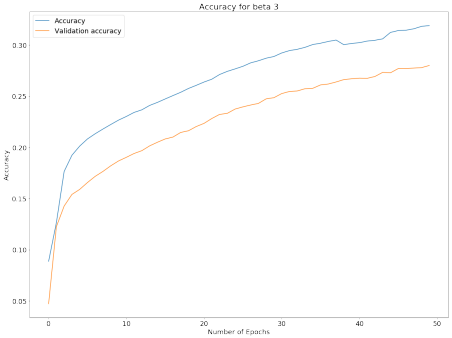


((c))

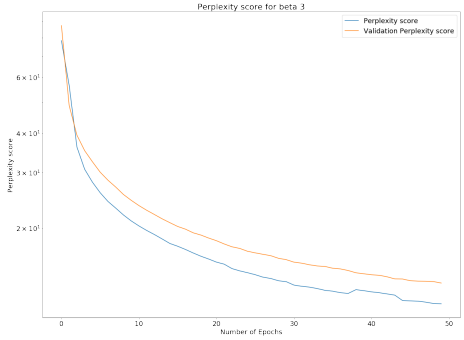


((d))

$\beta = 3$:

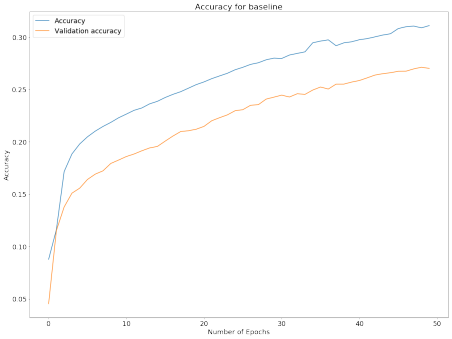


((e))

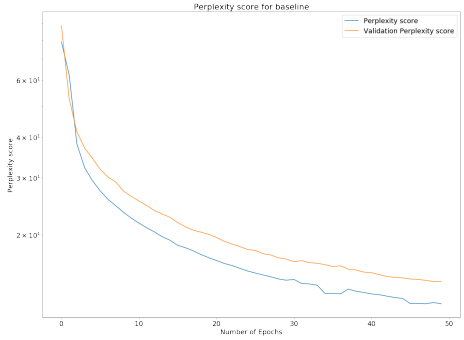


((f))

Baseline :



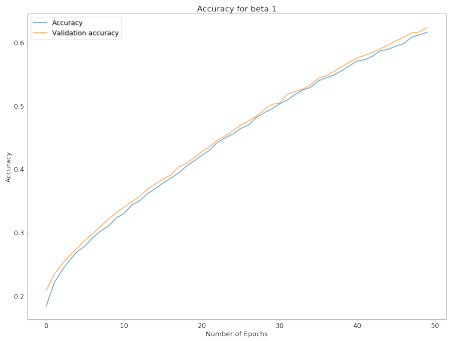
((g))



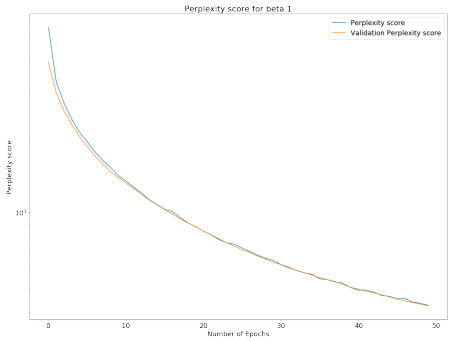
((h))

APPENDIX D
LEARNING CURVE IN MORE DETAILS: HOTEL DATASET

$\beta = 1$:

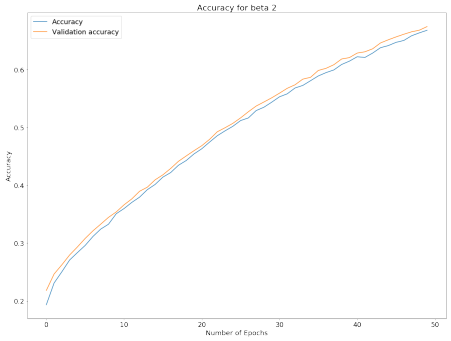


((a))

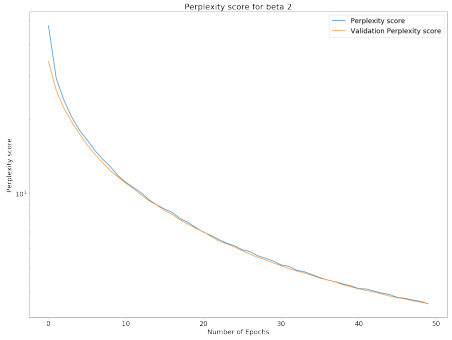


((b))

$\beta = 2$:

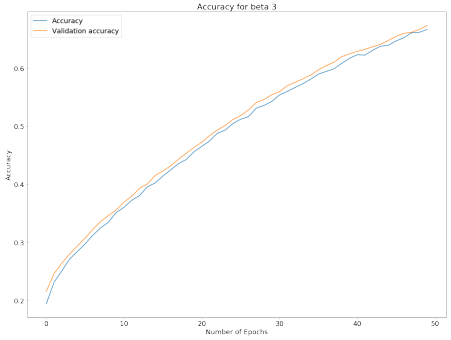


((c))

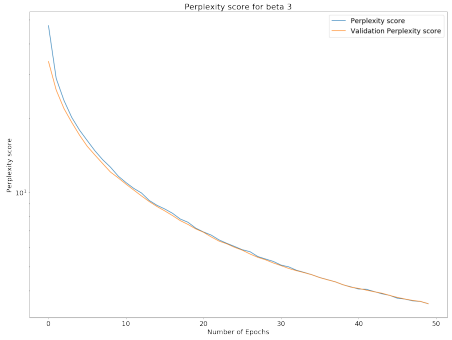


((d))

$\beta = 3$:

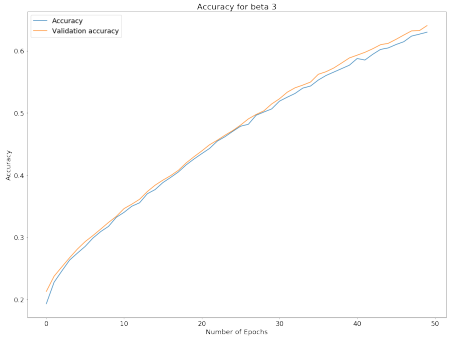


((e))

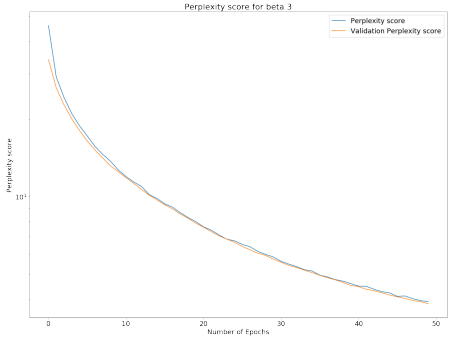


((f))

Baseline :



((g))



((h))

APPENDIX E
HUMAN TEST ANALYSIS: GRAMMATICAL CORRECTNESS

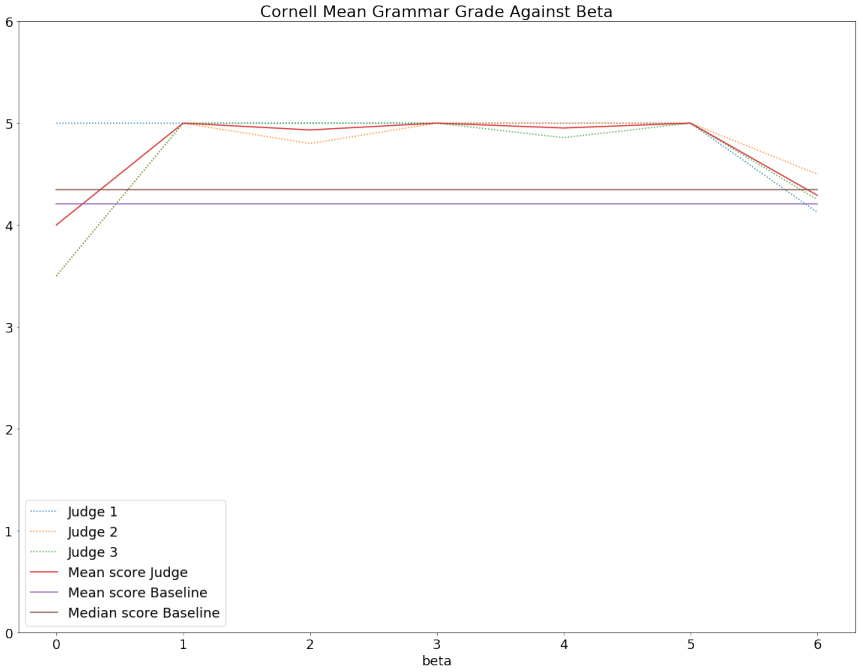


Fig. 8

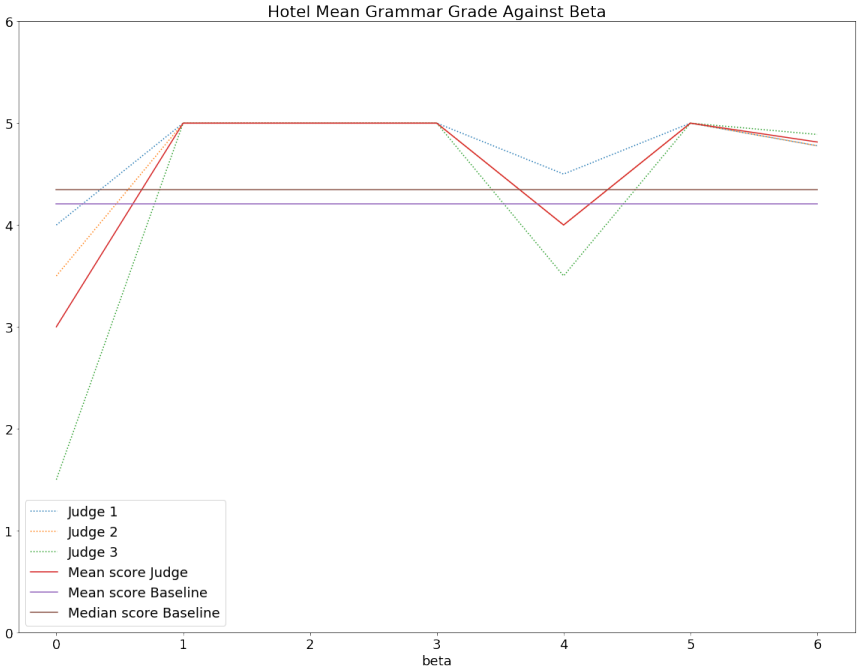


Fig. 9

APPENDIX F
HUMAN TEST ANALYSIS: AFFECT CATEGORIZATION

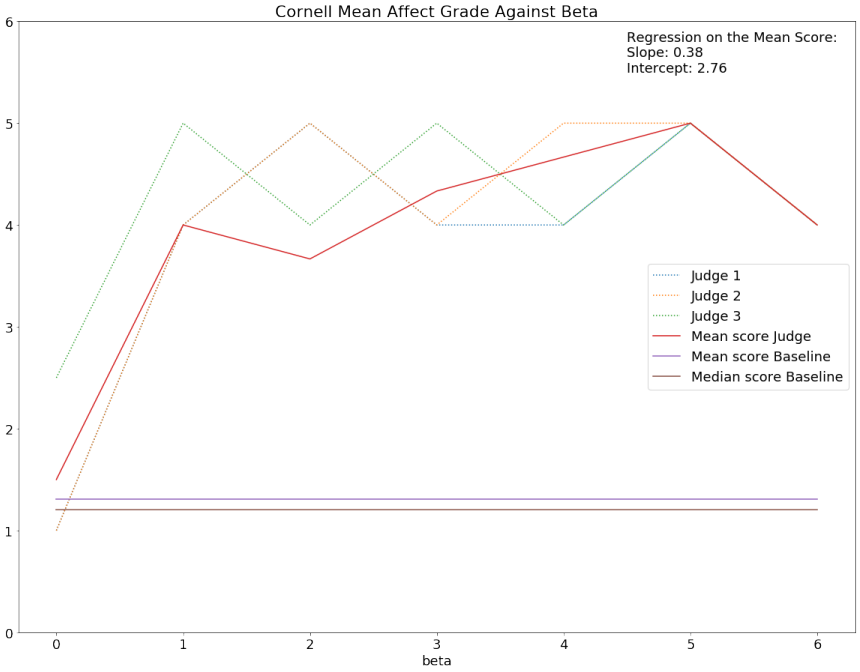


Fig. 10

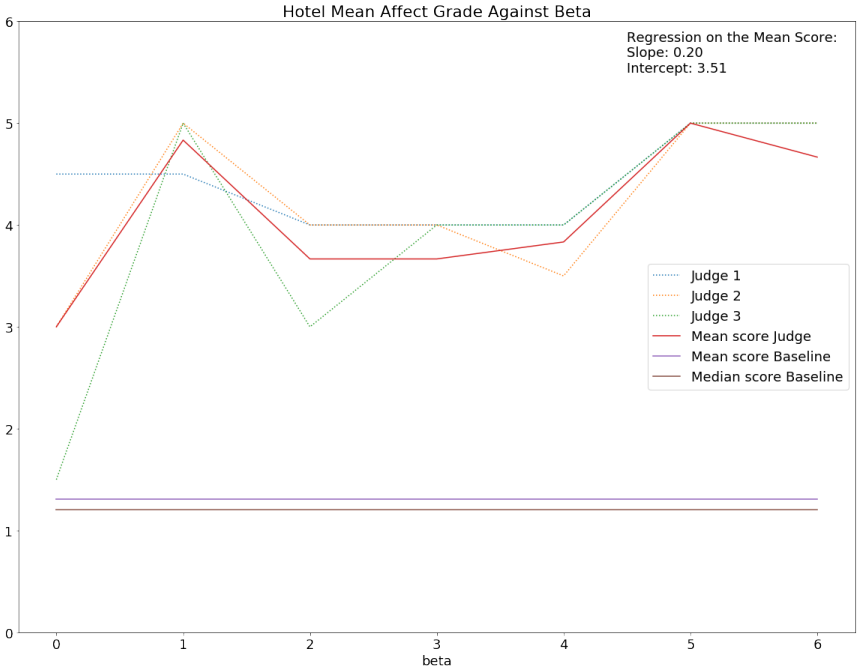


Fig. 11

APPENDIX G
HUMAN TEST ANALYSIS: AFFECT CATEGORIZATION (CONTINUED)

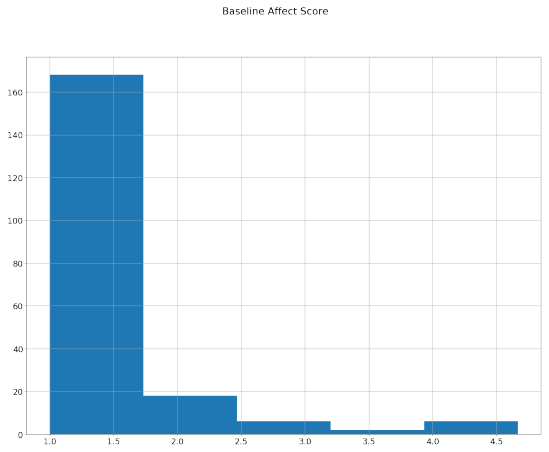


Fig. 12

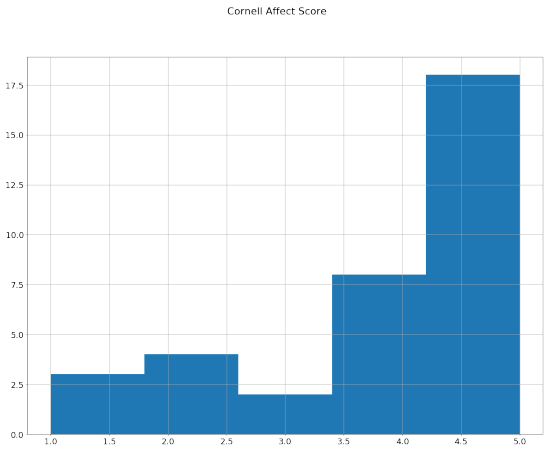


Fig. 13

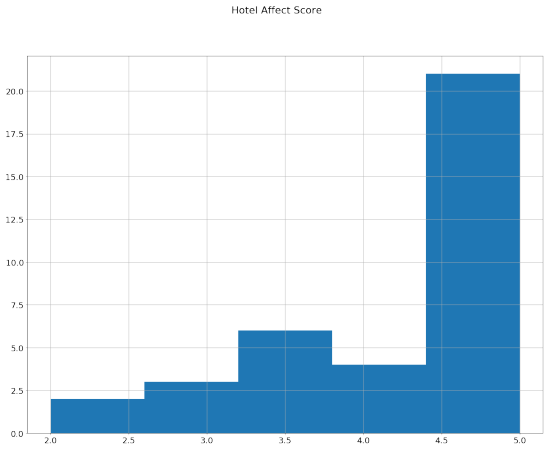


Fig. 14

APPENDIX H
SENTENCE GENERATION

Beginning	Sentence	Affect category	Beta
I feel like	i feel like a thief he doesn't like noise <End>	Positive Emotion	0
	i feel terrible about you to do something about me <End>	Sad	0
	i feel like a thief he doesn't like noise <End>	Anxious	0
	i feel like a thief he doesn't like noise <End>	Negative Emotion	0
	i feel like a thief he doesn't like noise <End>	Anger	0
	i feel better than that <End>	Positive Emotion	1
	i feel like a mutant <End>	Sad	1
	i feel guilty <End>	Anxious	1
	i feel terrible about you <End>	Negative Emotion	1
	i feel terrible about you <End>	Anger	1
	i feel good <End>	Positive Emotion	2
	i feel sorry for you <End>	Sad	2
	i feel guilty <End>	Anxious	2
	i feel terrible about you to ask me to come down to sleep <End>	Negative Emotion	2
	i feel like a mole in the middle of the street <End>	Anger	2
	i feel good <End>	Positive Emotion	3
	i feel sorry for you <End>	Sad	3
	i feel guilty <End>	Anxious	3
	i feel terrible what do you think <End>	Negative Emotion	3
	i feel like a teenager <End>	Anger	3
	i feel better <End>	Positive Emotion	4
	i feel sorry for you <End>	Sad	4
	i feel guilty <End>	Anxious	4
	i feel bad about it you know it <End>	Negative Emotion	4
	i feel like a teenager <End>	Anger	4
	i feel better <End>	Positive Emotion	5
	i feel sorry for you <End>	Sad	5
	i feel guilty <End>	Anxious	5
	i feel bad about you it not like you you know <End>	Negative Emotion	5
	i feel it <End>	Anger	5
	i feel like you <End>	Positive Emotion	6
	i feel that i just can't <End>	Sad	6
	i feel guilty <End>	Anxious	6
	i feel way <End>	Negative Emotion	6
	i feel it that <End>	Anger	6

TABLE VIII: Affect-LM on Cornell

Beginning	Sentence	Affect category	Beta
He was so	he was so lonely clair he knew he was dead <End>	Positive Emotion	0
	he was so lonely clair he knew he was dead <End>	Sad	0
	he was so lonely clair he knew he was dead <End>	Anxious	0
	he was so lonely clair he knew he was dead <End>	Negative Emotion	0
	he was so lonely clair he knew he was dead <End>	Anger	0
	he was so brave in his head he lives in the hospital to speak to the cia <End>	Positive Emotion	1
	he was so lonely <End>	Sad	1
	he was so lonely clair you knew he was dead <End>	Anxious	1
	he was so lonely clair you knew he was dead <End>	Negative Emotion	1
	he was so scared he was gone <End>	Anger	1
	he was so brave in the castle <End>	Positive Emotion	2
	he was so lonely <End>	Sad	2
	he was so lonely he had to work too much substantial for him <End>	Anxious	2
	he was so lonely <End>	Negative Emotion	2
	he was so scared he was a priest <End>	Anger	2
	he was so sweet <End>	Positive Emotion	3
	he was so lonely <End>	Sad	3
	he was so lonely clair you knew he was dead <End>	Anxious	3
	he was so lonely <End>	Negative Emotion	3
	he was so scared <End>	Anger	3
	he was so sweet <End>	Positive Emotion	4
	he was so lonely <End>	Sad	4
	he was so lonely he got a new gun <End>	Anxious	4
	he was so lonely <End>	Negative Emotion	4
	he was so scared <End>	Anger	4
	he was so sweet <End>	Positive Emotion	5
	he was so lonely <End>	Sad	5
	he was so lonely he got a new wheel <End>	Anxious	5
	he was so lonely <End>	Negative Emotion	5
	he was so intense <End>	Anger	5
	he was so good to me <End>	Positive Emotion	6
	he was so lonely <End>	Sad	6
	he was so lonely clair you knew he was a loser <End>	Anxious	6
	he was so bad bad <End>	Negative Emotion	6
	he was so intense <End>	Anger	6
I looked so	i looked so tired i was thinking about m at school where would you come back <End>	Positive Emotion	0
	i looked so tired i was thinking about m at school where would you come back <End>	Sad	0
	i looked so tired i was thinking about m at school where would you come back <End>	Anxious	0
	i looked so tired i was thinking about m at school where would you come back <End>	Negative Emotion	0
	i looked so tired i was thinking about m at school where would you come back <End>	Anger	0
	i looked so cute <End>	Positive Emotion	1
	i looked so much better than tears <End>	Sad	1
	i looked so much scared of the english justice <End>	Anxious	1
	i looked so much better than tears <End>	Negative Emotion	1
	i looked so embarrassed i didn't ask for that <End>	Anger	1
	i looked so cute <End>	Positive Emotion	2
	i looked so bad <End>	Sad	2
	i looked so much scared of the english <End>	Anxious	2
	i looked so much better than that <End>	Negative Emotion	2
	i looked so much scared of the old man <End>	Anger	2
	i looked so cute <End>	Positive Emotion	3
	i looked so bad <End>	Sad	3
	i looked so much scared of that stuff <End>	Anxious	3
	i looked so bad <End>	Negative Emotion	3
	i looked so much <End>	Anger	3
	i looked so impressed <End>	Positive Emotion	4
	i looked so bad <End>	Sad	4
	i looked so relaxed at the same time <End>	Anxious	4
	i looked so bad <End>	Negative Emotion	4
	i looked so embarrassed i was wrong <End>	Anger	4
	i looked so good <End>	Positive Emotion	5
	i looked so bad <End>	Sad	5
	i looked so relaxed at the same time <End>	Anxious	5
	i looked so bad <End>	Negative Emotion	5
	i looked so hateful <End>	Anger	5
	i looked so good <End>	Positive Emotion	6
	i looked so bad <End>	Sad	6
	i looked so relaxed <End>	Anxious	6
	i looked so bad <End>	Negative Emotion	6
	i looked so hateful <End>	Anger	6

TABLE IX: Affect-LM on Cornell (Continued)

Beginning	Sentence
I feel like	i feel like i didn't <End>
	i feel like i'm dead <End>
	i feel like i don't want to be a good boy <End>
	i feel like a bootlegger wife i don't like her <End>
	i feel like i don't know <End>
	i feel like you were in a band <End>
	i feel like i don't <End>
	i feel like that <End>
	i feel like i want to go to the bathroom <End>
	i feel like that <End>
	i feel like you don't even know <End>
	i feel like i were in the cafeteria today <End>
	i feel so good <End>
	i feel like he wouldn't know <End>
	i feel like i don't know <End>
He was so	he was so nice derek <End>
	he was so good on him <End>
	he was so lucky about him he married you before not <End>
	he was so mail to be a duck <End>
	he was so clean <End>
	he was so impressed with your mother <End>
	he was so <End>
	he was so either he <End>
	he was so nervous <End>
	he was so scared in time he probably likes to know what happens to him <End>
	he was so scared <End>
	he was so excited he could want to know how he kills school <End>
	he was so in the closet <End>
	he was so tall <End>
	he was so beautiful he been havin a lot of fun <End>
I looked so	i looked so hard i was a little bit scared about that <End>
	i looked so i was just thinking about it <End>
	i looked so i didn't choose you <End>
	i looked so i would not know the difference <End>
	i looked so early <End>
	i looked so far apart you <End>
	i looked so i ought to know <End>
	i looked so i could get a feeling that all <End>
	i looked so became a bit of a great deal of stress you know it a new place <End>
	i looked so early <End>
	i looked so far away <End>
	i looked so i do not know how to get mixed up in it <End>
	i looked so hard you could have told you that <End>
	i looked so far apart the same <End>
	i looked so hard on the whole day <End>

TABLE X: Baseline on Cornell

Beginning	Sentence	Affect category	Beta
The hotel was	the hotel was terrible and deep in the bathroom <End>	Positive Emotion	0
	the hotel was terrible and deep in the bathroom <End>	Sad	0
	the hotel was terrible and deep in the bathroom <End>	Anxious	0
	the hotel was terrible and deep in the bathroom <End>	Negative Emotion	0
	the hotel was terrible and deep in the bathroom <End>	Anger	0
	the hotel was very nice and helpful <End>	Positive Emotion	1
	the hotel was filthy <End>	Sad	1
	the hotel was very nice <End>	Anxious	1
	the hotel was filthy and disgusting <End>	Negative Emotion	1
	the hotel was very dirty <End>	Anger	1
	the hotel was very nice and helpful <End>	Positive Emotion	2
	the hotel was not very nice and helpful <End>	Sad	2
	the hotel was under construction <End>	Anxious	2
	the hotel was filthy and disgusting <End>	Negative Emotion	2
	the hotel was obviously rude <End>	Anger	2
	the hotel was very nice and clean <End>	Positive Emotion	3
	the hotel was not very nice and the staff was rude <End>	Sad	3
	the hotel was not very clean and the area was absolutely horrible <End>	Anxious	3
	the hotel was filthy and not very helpful <End>	Negative Emotion	3
	the hotel was not very clean and the staff was very rude and ignorant <End>	Anger	3
	the hotel was very clean <End>	Positive Emotion	4
	the hotel was not very nice and the room was nice <End>	Sad	4
	the hotel was not very clean and the area was very dirty <End>	Anxious	4
	the hotel was not very nice and the room was very bad <End>	Negative Emotion	4
	the hotel was not very clean and the staff was very rude and ignorant <End>	Anger	4
	the hotel was not very clean and the area was very nice <End>	Positive Emotion	5
	the hotel was not very nice and the room was not very nice <End>	Sad	5
	the hotel was not very clean and the area was and the lobby was disappointing <End>	Anxious	5
	the hotel was not very small and construction <End>	Negative Emotion	5
	the hotel was not very clean and the staff was very rude and ignorant <End>	Anger	5
	the hotel was not very clean and the area was and the lobby was comfortable <End>	Positive Emotion	6
	the hotel was not the quality of the hotel <End>	Sad	6
	the hotel was not very clean and the area was and the beds were not comfortable <End>	Anxious	6
	the hotel was very bad <End>	Negative Emotion	6
	the hotel was not very clean and the staff was very rude and ignorant <End>	Anger	6
The place was	the place was terrible <End>	Positive Emotion	0
	the place was terrible <End>	Sad	0
	the place was terrible <End>	Anxious	0
	the place was terrible <End>	Negative Emotion	0
	the place was terrible <End>	Anger	0
	the place was terrible <End>	Positive Emotion	1
	the place was dirty and stiff <End>	Sad	1
	the place was terrible <End>	Anxious	1
	the place was terrible <End>	Negative Emotion	1
	the place was terrible <End>	Anger	1
	the place was very comfortable <End>	Positive Emotion	2
	the place was dirty and stiff <End>	Sad	2
	the place was very uncomfortable <End>	Anxious	2
	the place was terrible <End>	Negative Emotion	2
	the place was very uncomfortable <End>	Anger	2
	the place was very nice and the staff was friendly <End>	Positive Emotion	3
	the place was very uncomfortable <End>	Sad	3
	the place was terrible and the air conditioning and the desk was very dirty <End>	Anxious	3
	the place was filthy and not very clean <End>	Negative Emotion	3
	the place was very uncomfortable <End>	Anger	3
	the place was very comfortable <End>	Positive Emotion	4
	the place was very uncomfortable <End>	Sad	4
	the place was very comfortable and uncomfortable <End>	Anxious	4
	the place was filthy and not very helpful <End>	Negative Emotion	4
	the place was very uncomfortable <End>	Anger	4
	the place was very comfortable <End>	Positive Emotion	5
	the place was not the but of the service <End>	Sad	5
	the place was very comfortable <End>	Anxious	5
	the place was very dirty <End>	Negative Emotion	5
	the place was very tiny and the staff was dirty <End>	Anger	5
	the place was very comfortable and the beds were comfortable and the room was not very clean <End>	Positive Emotion	6
	the place was not the but it was a bad cafeteria <End>	Sad	6
	the place was very uncomfortable <End>	Anxious	6
	the place was very bad and the smell of the beds i have to do was the update in the middle of the course of the hotel <End>	Negative Emotion	6
	the place was stained and the pool was not the time <End>	Anger	6

TABLE XI: Affect-LM on hotel