An Emotion-Enriched Sentence-Edit Natural Language Model (ES-NLM)

CS 224U Final Report

Sharman Tan
sharmant@stanford.edu

Nitya Mani
nityam@stanford.edu

Jaewoo Jang
jaewooj@stanford.edu

1. Introduction

Chatbots and intelligent personal assistants (IPAs) surround us. Many homes and smartphones have a nearby Alexa, Google Voice Assistant, Siri, or Cortana on hand to help set alarms, answer questions, and create smoother interactions with technology. The ultimate goal of such IPAs is to at least partially replace human assistants such that humans can rely on technology when it comes to repetitive, concurrent, or complex tasks. As a cost-cutting measure and harbinger of the future, chatbots have been adopted by almost every industry to partially replace the need for human customer service agents and/or call centers. Even if they are not visually apparent, chatbots are omnipresent in the current workings of society, from handling bank websites to managing customer complaints to booking flights and rides on apps.

However, despite the prevalence of IPAs, many users express frustration when communicating with them. The current state of IPAs is such that assistants are generally quite capable and robust when it comes to answering structured or common questions. They function mostly like machines in society but have progressively been programmed to sound more natural and human-like. They give off friendlier vibes that might suggest they are something more than just pieces of machinery; however, the problem lies with the fact that these IPAs have not developed emotional intelligence at their cores. No matter how many jokes or colloquial phrases these IPAs are programmed to use, it does not take long for users to spot the gap between intelligent personal assistants and *emotionally* intelligent personal assistants.

User likeability and engagement are arguably the two most important factors when creating a successful IPA. How users evaluate the empathetic capabilities of IPAs directly correlates with how they perceive the overall performance and quality of the IPAs. To optimize this relationship, we must focus on developing the "human-like" aspects of IPAs from their cores.

While a tremendous amount of research has been published in the last several years using both shallow and deep learning methods to enhance the information capture and delivery of natural language generation-based virtual assistants, much less focus has been on imbuing current assistants with convincing emotional or affective capabilities.

Here, we present a generative model to improve the conversational capabilities of IPAs using an emotion enhanced edit-based model that creates plausible, diverse sentences with an emphasis on a specified emotional state. We revisit some of the recent literature in Section 2 and highlight our overall approach and model architectures in Sections 3 and Section 4, respectively. In Section 5, we describe the dataset we train and evaluate our models on, and in Section 6, we specify the particular qualitative and quantitative measures we rely on when evaluating model performance. Finally, we present and discuss our findings on the performance of our models in Section 7.

2. Previous Work

Natural language generation (NLG) is an extremely challenging problem with a history of machine learning approaches throughout published research. We highlight a few of the most relevant and recent works below, but many other models (e.g. deep learning models) have had some efficacy in generating plausible natural language or emotional responses.

2.1. Context-Aware, Plausible NLG

Google's *A Neural Conversation Model* (12) was one of the first papers to use a Long-Short Term Memory (LSTM) encoder-decoder *seq2seq* framework to train a model that responds to a given query. (12) designed a conversational agent to return a response that best suits the context and the given query. After training the model on an IT helpdesk troubleshooting dataset and a movie subtitles dataset, the model was able to naturally and smoothly handle conversation. However, it lacked consistency, particularly when the topic domain changed.

Several follow-up works focused on generating plausible textual responses to user queries, largely using a left-to-right from-scratch approach. While this approach may be ideal when it comes to generating completely original responses from scratch, it poses difficult challenges in terms of building fully grammatical sentences, given the complexity and depth of language. Therefore, instead of focusing

on NLG in terms of building sentences completely from scratch, Guu et. al. introduce a new generative language model for building sentences from existing prototypes. In their paper *Generating Sentences by Editing Prototypes* (6), they create a *prototype-then-edit* model that samples a prototype sentence from a corpus and randomly picks a neural editor ("an edit vector"). The model then generates a new sentence, conditioned on both the prototype sentence and the edit vector. Because the sentences are not generated anew but are rather lexically and semantically edited versions of the prototype sentence, this approach widens the diversity of sentence generation while guaranteeing grammatical consistency with much less difficulty. The prototype-then-edit model circumvents the need to generate new sentences altogether. Although this may seem to narrow the scope of generated sentences, with enough edit vectors, this approach can produce a vast array of sentences for any given prototype sentence. Given the complexity of language models in general and the prevalence of natural language datasets, this prototype-then-edit model is extremely promising in the context of diversifying the language capabilities of IPAs. Therefore, we base our text generation approach off of a variant of this model, focusing not only on how to diversify generated sentences, but also on how we can encode emotional information in them.

2.2. Emotional NLG

While many NLG papers have focused on improving the quality of content generation, few works have considered the question of incorporating emotional intelligence into IPAs and content generators. In *Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory* (16), the authors generate an emotional response given an emotional state and a query. They use the Emotional Chatting Machine (ECM), an encoder and decoder 2-layer GRU, that learns the vector representation of each emotion and inputs each latent representation into the decoder. The paper highlights how chatbots' responses can change for identical inputs given different emotional states. However, the model suffers from a high degree of randomness and lack of plausibility in the responses.

Ghosh et al. proposed Affect-LM (4), an extension to LSTM-based language generation models, to generate conversational text while conditioning on affect categories. The model allows various degrees of emotional expression in the generated text based on a tunable design parameter. Ghosh et al. infer an affect category from the context data and define a feature extractor to detect keywords in text and infer emotion in the context. Affect-LM predicts each next word conditioned not only on previous words, but also on this affect category.

3. Approach

We work to improve the conversational capabilities of IPAs by focusing on their emotive capabilities rather than strictly content-based capabilities and by taking a prototype-then-edit (6) approach rather than generating textual responses from scratch. Our efforts focus on dynamic text modification of an existing corpus to re-purpose existing textual responses into stronger, more relevant ones. We seek to combine the best of the two groups of models described above by generating high quality, plausible, and context-aware textual responses that additionally include intelligently chosen emotional inflections.

Specifically, we implement a deep learning NLG model that modifies a prototype sentence based on a specified emotional state (e.g. positive or negative). This approach differs from more mainstream methods used to generate textual responses in chat environments, which mainly involve recurrent neural language models (NLMs) that generate sentences from scratch. NLMs often struggle to balance the trade-off between grammatical correctness and diversity of utterances, resulting in unintelligible, implausible, or narrowly repetitive responses that render chatbots' communications as robotic, thus undermining the user-bot connection.

We leverage a variant of the prototype-then-edit model first introduced by Guu et al. (6) that samples a random prototype sentence from the training corpus, randomly draws an "edit vector" from a prior distribution, and applies a neural editor to generate a new sentence. We call our model an emotion-enriched sentence-edit natural language model (ES-NLM) and provide details on our model design in later sections. We adopt this approach to modify prototype sentences while additionally conditioning on emotion-enriched latent edit vectors, generated via a process similar to that in (1) to learn emotion enriched word representation.

We hypothesize that we can dynamically adapt existing text to encode further emotional information based on a provided context. More specifically, we hypothesize that we can leverage a prototype-then-edit based deep learning model and existing natural language processing based low-dimensional embedding methods to modify existing sentences according to emphasize desired positive or negative emotions.

4. Models

4.1. Baseline Model

Our baseline model is the variational autoencoder (VAE) from *Generating Sentences from a Continuous Space* (3). Standard VAE models (7) consist of an encoder and decoder and represent observations in latent space in a probabilistic manner. (3) adopts the standard VAE model where the the continuous latent variable z is used to capture latent infor-

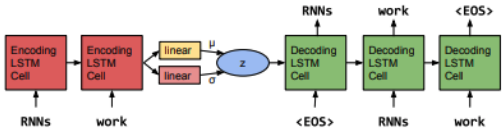

Figure 1. **Baseline** VAE model that naively learns to decode an encoded representation. The model is forced to learn all the plausible decoded sentences from every point in the latent space based on some reasonable Gaussian prior (14).

mation from the data. Our main goal in using a simple continuous latent variable in this paper is to capture the global information and complex representations of language models. While a naive autoencoder and decoder architecture is limited to capturing the local context of the encoded input, the standard VAE introduces the continuous latent variable representing the maximum likelihood of a sample x, conditioned on this latent variable. Therefore, encoding x and decoding this encoded representation is no longer a deterministic process, as we may sample a different value of the latent variable z each time, unlike our baseline model, which samples this latent variable uniformly.

Authors of this baseline model (3) propose that these continuous latent variables model the stylistic, topical, and overall high-level syntactic features of sentences. Therefore, VAEs learn these latent representations not as static points, but rather as ellipsoidal regions of a latent space. Each transformation of each data point is not learned in isolation, but rather depends on other mappings within this latent space.

The baseline autoencoder-decoder VAE model naively maps its prototype sentence to some encoded vector space. Then, this region in the latent space transforms into a new latent space, conditioned on a single continuous latent variable z, and the decoder recovers the word from this encoded representation. The baseline model is trained using *NLTK Corpus* (8) packages including Reuters, Gutenberg, Brown and Wordnet corpus. Because this baseline VAE model attempts to learn the higher level features of the English text, it must be trained on a stream of connected sentences, rather than on snippets of sentences such as in the Yelp Review dataset.

This baseline model is an adaption of the *Toni Antonova* github repository (2)

The approach of directing a sample sentence to some new representation is analogous to the prototype-then-model. Our ES-NLM adopts the same base model of a VAE. (6) differs from the naive implementation in that (6) uses a von Mises Fisher (vMF) distribution in their prototype-then-edit model(14), motivating our selection of this vMF VAE in our ES-NLM model as well. Rather than computing a Gaussian distribution over multi-dimensional latent space, our model focuses on sampling the latent vector z over some dense $p-1$ dimensional sphere that fits the n-dimensional vector space representation(14). Compared to the baseline model, adopting vMF distribution theoretically imposes two main benefits: 1). The input and the output texts of the model will lie very close to each other in the latent space 2). with non-diverse dataset, capturing the complexity of language model with the naive baseline model is very difficult. The vMF distribution ensures that each latent vector covers only a small region over a multi-dimensional latent space. (14)

Therefore, we implement our baseline model with the following default hyperparameter settings:

- Batch size: 500
- Pre-trained Word2vec: wiki.en.env
- Optimizer: RMSProp
- Epochs: 20
- Learning rate: 0.001

4.2. ES-NLM

As mentioned above, we use an ensemble of 1) a semi-supervised model with a prototype-then-edit based unsupervised approach to generate candidate sentences from randomly sampled sentences from our training corpus, and 2) an additional sentiment analysis based approach to classify and select the optimal candidate sentences corresponding to the desired emotional state. Overall, we refer to this model as the *emotion-enriched sentence-edit natural language model* (ES-NLM).

The bulk of our work rests on sentence generation not from-scratch, but rather by randomly sampling and editing sentences from our training corpus via a neural editor. Figure 9 highlights this modeling approach at a high level.

Ultimately our goal is to maximize the log likelihood of an edited sentence $p(x)$ which we lower bound using the conditional probability of generation from prototype sentences from the training set x' with small Jaccard distance from x as

$$\log p(x) \geq |\mathscr{N}(x)|^{-1} \sum_{x' \in \mathscr{N}(x)} \log p(x \mid x') + \frac{\log |\mathscr{N}(x)|}{|\mathscr{X}|}$$

where $\mathscr{X}$ is the training set, and

$$\mathscr{N}(x) = \{x' \in \mathscr{X} \mid d_J(x,x') < 0.5\}$$

are those prototypes x' that have high lexical overlap with x measured by a Jaccard distance smaller than 0.5. Unfortunately,

$$\log p(x \mid x') = \log \mathbb{E}_{z \sim p(z)}[p_{edit}(x \mid x', z)]$$

is impossible to easily directly approximate as the expectation over the distribution of edit vectors $p(z)$ has no closed

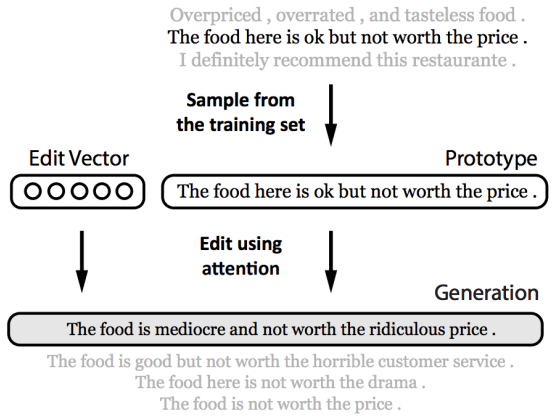

Figure 2. At a high level, the Prototype-then-Edit model generates a sentence or phrase by first drawing a random training example that it then edits using an edit vector drawn randomly from a prior distribution (6)

Example	Transformed example
It is important to remain watchful.	It is important to remain watchful.
It is important to remain watchful.	To remain watchful is important.
They announced that the president will restructure the division.	The president will restructure the division, they announced.
We should march because winter is coming.	Winter is coming, because of this, we should march.
The best restaurant of New York.	New York's best restaurant.
The talk was denied by the boycott group spokesman.	The boycott group spokesman denied the talk.
It is important to remain watchful.	It remain is to important watchful.

Figure 3. ES-NLM without emotional classifier. This figure demonstrates how well the sentence edits perform. Qualitatively, the edited sentences are clearly much different from the prototype sentence. We hope to qualitatively assess how likely these transformed examples reflects positive or negative sentiment.

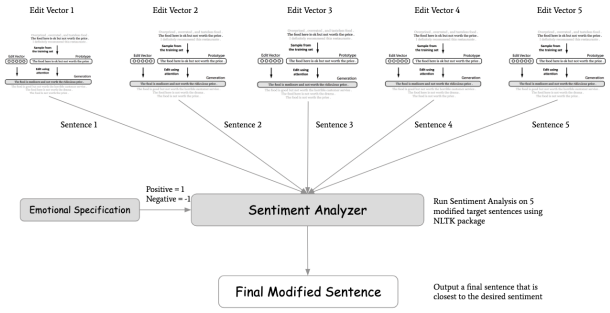

Figure 4. ES-NLM model learns to choose the latent variable that maximizes a certain emotion specification (positive or negative) of the edit. The sentiment analyzer uses the NLTK Sentiment Analyzer package (8) to evaluate the score.

form and approximating via Monte Carlo sampling has high variance since $p_{edit}(x \mid x', z)$ is zero for almost all z. Thus we approximate with a lower bound given by taking the expectation of $\log p_{edit}(x \mid x', z)$ over a distribution $q(z \mid x', x')$ adjusted by the KL divergence between q and $p(z)$ that we learn using a variational autoencoder (q) and decoder (p_{edit}) to obtain the approximation of the log-likelihood $\log p(x)$ of our obtained sentence x that we actually optimize via stochastic gradient ascent. We describe the four major pieces of our generative model in more detail. The first three pieces and the above approximation are derived from the model in (6).

4.3. Neural editor $p_{edit}(x \mid x', z)$

We use a sequence-to-sequence (seq2seq) attention-based model where the input is a prototype sentence x' from the training dataset and the output is our edited sentence x based on a perturbation from a supplied edit vector z. This model uses an encoder-decoder architecture (13) where both encoder and decoder are 3-layer bidirectional LSTMs. The encoder takes as input word vectors that we initialize using GloVe (10). All other layers take as input a concatenation of the forward and backwards states (hidden) of the previous layer, as in (10).

The decoder uses an attention network 3-layer LSTM, concatenating an attention context vector with the top-layer hidden state. Ultimately we use softmax to compute a distribution over output tokens.

4.4. Edit prior $p(z)$

We sample an edit vector $z \sim \mathscr{D}$ from a prior distribution by first uniformly at random sampling a length $||z||_2 \sim \text{Unif}(0, 10)$ and subsequently sampling a uniform random direction on the unit sphere.

4.5. Sentiment Analysis.

Quantitatively, we also measure the positive or negative sentiment of the newly generated sentences by evaluating them on the sentiment analysis package offered by `NLTK` package (8). By using its sentiment analysis module, we evaluate the sentiment strength of the sentence where positive and negative floating points denote positive and negative valence, respectively. This measures how well the semi-supervised classifier performs in finding the right edit vector in generating a sentence based on the emotional specification.

Although our model relies on sentiment analysis to optimize the proximity of generated sentences to the given emotional specification, our results are dependent on the accuracy of this sentiment analysis tool. Therefore, to evaluate how emotionally expressive our generated sentences are, we turn to human-based evaluation.

4.6. Inverse neural editor $q(z \mid x', x)$

We use a neural editor to approximate what edit vectors z are likely to produce edited sentence x from the prototype sentence x'. This probability is generated via a generalization of the intuition that if x', x differ by a single word then the best possible z should the vector corresponding to that differing word. More generally we construct edit vectors as the sum of word vectors in the difference between x' and x (either inserted or deleted) by concatenating the corresponding word embeddings initialized via GloVe. We then add a random perturbation via von-Mises Fisher (vMF) noise and uniformly perturb the magnitude of the corresponding vector.

This approach to choosing a distribution for q is slightly different from the common distributional choice in a variational decoder. One advantage of this approach is that it allows the log likelihood to decay with cosine similarity, which naturally captures the fact that our edit vectors are constructed from perturbed concatenated word vectors.

4.7. Emotional enrichment

Using a list of high likelihood edit vectors, we secondarily optimize over emotional state in a supervised learning step that uses a labeling associated to training sentences and the NLTK sentiment analysis tools (8) to label candidate edited sentences by the associated conveyed sentiment and select the best matching candidate. This enrichment approach is inspired by work of (1) with the goal of maintaining high quality sentences as we introduce an emotional angle. The goal of such a model is to emphasize the existing emotion in an edited sentence.

5. Data

We evaluate our models quantitatively and qualitatively primarily on the Yelp review corpus (15). The Yelp restaurant review corpus was provided to users as part of the Yelp dataset challenge. The Yelp dataset is a subset of Yelp businesses, reviews, and user data, giving in JSON format. It includes 6,685,900 Yelp recommended reviews from 192,609 businesses in 10 metropolitan areas. Each review contains the full review text data, the user ID of the person that wrote the review, and the business ID that the review is written for. Importantly, it also includes the number of stars the reviewer gave as well as how many votes the review received for being useful, funny, or cool.

We preprocess this data by replacing named entities as identified by the spaCy named-entity recognizer (NER) with their associated categories, and further replace tokens outside of the top 10,000 most frequent words with a single token to indicate rarity and exclusion from our limited vocabulary.

We use these reviews as the baseline text for a generative model that samples a sentence from our training set and augments and/or edits the text to create a new sentence, building on work of Guu et al. (6). (6) uses the full Yelp dataset and trains for 400,000 iterations. However, this spanned nearly a week of training, likely using GPUs. Due to time and resource constraints, we use approximately 10,000 reviews, dividing this randomly subsampled dataset into training, validation, and test sets with a ratio of 60%/20%/20%. We then run our model for 5,000 iterations on CPUs for about 11 hours.

6. Metrics

We evaluate our model's results using a combination of qualitative, quantitative (model-based), and quantitative (human-based) metrics.

6.1. Quantitative

We use standard quantitative measurements to impute the log likelihood of generating our sentence and to measure the semantic similarity between our generated sentence and the prototpy; this allows us toe impute the plausibility and thus "human-like" quality of the text we generated. We use the BLEU score described in detail below to compare our baseline VAE model to the ES-NLM and measure training and validation loss. To directly measure the sentence quality, we use a survey to ask human participants to rank sentences based on their likelihood to be written by a happy/sad human.

BLEU. To measure fluency and sentence quality, we measure perplexity and calculate the Bilingual Evaluation Understudy Score (BLEU) (9). Like (6), we use perplexity to measure improvements on language modeling. BLEU is a metric for evaluating a generated sentence to a reference sentence, where a perfect match corresponds to a score of 1.0 and a perfect mismatch results in a score of 0.0 (9). Therefore, we use the prototype sentence as the reference sentence and compare the sentences generated using edit vectors to that reference sentence using BLEU.

Human-based. We surveyed 12 Stanford students, all of whom speak, read, and write English with a high level of fluency. In the online survey, we list five sets of five sentences, for a total of 25 sentences generated by our model. Each set of five sentences corresponds to five sentences generated for one prototype sentence, using different latent variable z. The directions for the survey are as follows: "For each of the following sets of sentence snippets, rank them from 1 (best/most likely) to 5 (worst/least likely) based on the given question." For three of the five sets of sentences, the accompanying question is "How likely are these responses to have been written by a *happy* Yelp reviewer?" and for the remaining two sets of sentences, the accompanying question is "How likely are these responses to have been written by an *unhappy* Yelp reviewer?" We

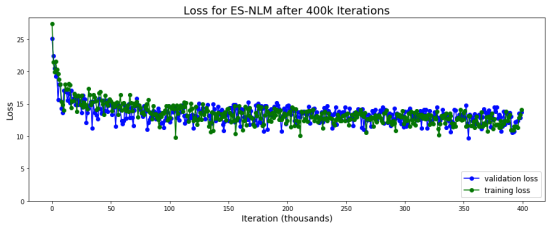

Figure 5. The training and validation loss graph over 400k iterations

then analyze these responses.

7. Results

7.1. Training and Validation Loss

One quantitative metric we used to track the learning trend of each model was its loss value after each 500 iteration. Due to computation constraints, we were only able to run a 10k training sample on Coda lab after 5000 iterations, but we have obtained a saved checkpoint from Guu et al.'s Codalab that trained on the same dataset except over all 6,685,900 Yelp recommended reviews and over 400,000 iterations. Tracking the loss was an important metric for evaluating whether the model was learning the various latent variable z. By verifying that the model was indeed learning, we ensure that our model is well-trained on our particular dataset and also ensure that the qualitative assessment of the output solely measures the effectiveness of the model.

As seen in Figure 4, the loss of the model converges before 2,500th iterations until both its training and validation loss go into a plateau. Though it may not be noticeable over a couple of iteration data points, the general loss value is still converging to 0.20. Since our ES-NLM model should learn to encode-decode a general set of source sentence to a general set of target sentence within the same latent space, we did not quite expect the loss function to fully converge to a flat 0.

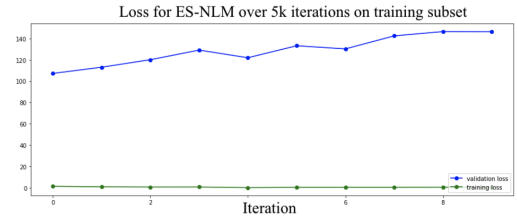

Figure 6. The training and validation loss graph after 5k iteration over smaller training subset.

After directly running our ES-NLM for 5,000 iterations over a much smaller subset of the training data, we notice that the model does not quite learn to generalize the latent variable z to the validation set. There were merely 1,000 training and validation samples. The fact that there were only 1000 possible source and possible target sentences indicates that this subset is no where representative of the English Language Model. Therefore, despite the training loss being close to 0, this model overfits and thus does not quite learn to reconstruct any unseen sentences from the validation set.

7.2. BLEU Score

Evaluating text generation is a difficult, intricate task. As mentioned in the metrics section, BLEU score evaluates how the generated text matches the original sentence. One key assumptions with using BLEU score is that *lexical similarity* denotes *semantic similarity* (11). By computing n-gram overlap between the source and the target sentence, we assume that this overlap signifies similarity, which in most context is true.

Based on Figure 6, single data point over each iteration is quite arbitrary as it depends on how the model determines the source and the target sentences (i.e., if the target sentence was chosen to be less "matched" than the source, then on that iteration, the model is bound to have a low BLEU score). However, similarly to its loss score, there is a general upward trend in the BLEU score as well. Since our model aims to edit sentences, our model does not look for complete reconstruction of its original sentence but it attempts to make reasonable amount of edits. It initializes with 0.2 BLEU score edits to generating texts, that score roughly 0.4-0.5 on the BLEU score scale. To put this value in context,

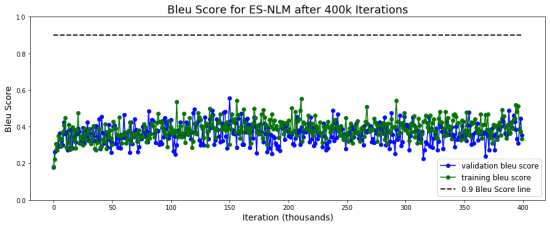

Figure 7. The BLEU score of the model after 400k iteration over the full training dataset

Running the same model for 10k iterations and with smaller training subset does not yield an increasing BLEU score for the validation set. Training the model on the full dataset suggests that some sentences in the training set resemble sentences in the validation set, and therefore, editing source sentences in the validation set turns out to "match" well with the source. From figure 9, the ES-NLM model does a more effective job maintaining lexical similarity between the source and the output sentences. Also, the average BLEU score between the training and the validation test set is lower with ES-NLM than with the baseline. This

BLEU Score	Interpretation
< 10	Almost useless
10 - 19	Hard to get the gist
20 - 29	The gist is clear, but has significant grammatical errors
30 - 40	Understandable to good translations
40 - 50	High quality translations
50 - 60	Very high quality, adequate, and fluent translations
> 60	Quality often better than human

Figure 8. The BLEU score for ES-NLM after 400k iteration achieves 0.4-0.5 BLEU score which is relatively a good edit since our model does not aim to completely recover the same source sentence as its output (11)

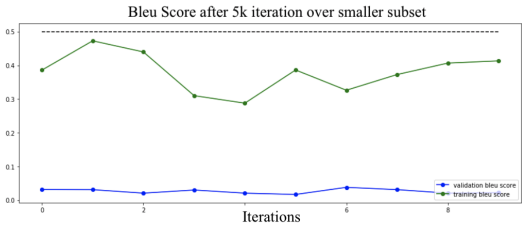

Figure 9. The BLEU score of ES-NLM after 10k iteration over a subset of the training set. The BLEU score for the sentences in the validation set does not increase after each training because the sparse training sample does not reflect the sentences in the validation set

indicates that ”editing” sentences is more effective in recovering the original sentence while generating matching sentences left-to-right using the VAE model is more difficult. Also, for both models, the training and the validation test set were derived from the same corpus, meaning that the characteristics of both samples do not highly differ. We find that using the vMF distribution captures this language model similarity in that ES-NLM achieves only 0.03 BLEU score difference between the two sets. The VAE model on the other hand generates widely differing texts even though the sources appear lexically similar.

Model	BLEU Score (Train)	BLEU Score (Test)
Baseline	0.271	0.192
ES-NLM	0.429	**0.392**

Table 1. The BLEU score of ES-NLM trained on the full dataset (for 400k iterations) and baseline VAE model, averaged over the last 10 iterations. The BLEU score of the ES-NLM was higher than that of the baseline VAE model, which meant that the similarity of the source and the output sentence was higher.

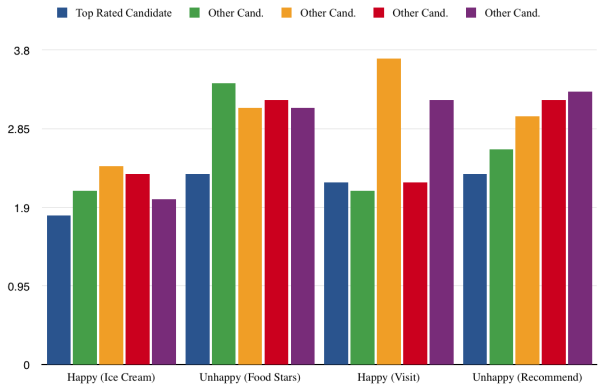

Figure 10. We tested five candidate sentences in each of four scenarios generated by our generative models with a survey audience and asked them to rank how likely from a scale of 1 (best/most likely) to 5 (worst/least likely) they would be to be part of a happy/sad human Yelp review. One of these candidates was the sentence we selected, and we present the average rankings

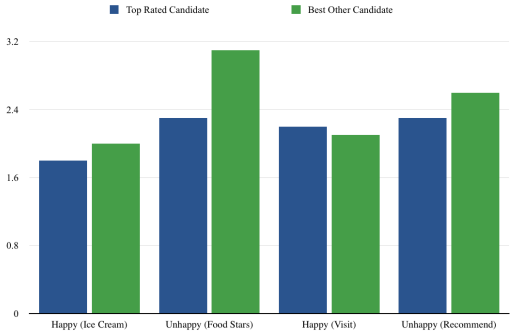

Figure 11. For clarity, we directly compare our generated sentence to whichever of the other four candidates was deemed the most likely and chart the average rankings they received.

7.3. Survey Results

Based on Figure 11 and 10, sentences generated and selected by the ES-NLM achieved the most likely score (a number closer to 1) of being generated by a human Yelp reviewer with the specified emotional state. The dark blue bars in figure 10 was the sentence edited by ES-NLM and survey participants thought these sentences were most emotionally-enriched. This verifies that our model was able to select and train the most appropriate latent variable z to edit the source sentence to a new sentence that is semantically consistent and emotionally enriched.

7.4. Qualitative

In Figure 12, we sample three source sentences and their corresponding outputs. The output sentences of ES-NLM

	Source Sentence	Generated Sentence
Baseline VAE	this food was amazing one of the best I've tried, service was fast and great	this is the best <norp> food in <gpe>
	the coffee ice cream was one of the best I've ever tried	the food was great, but not great
	<time> , another server came to take order .	The next server came at <time>
ES-NLM	this food was amazing one of the best I've tried, service was fast and great	the food was delicious, and the service was very friendly
	the coffee ice cream was one of the best I've ever tried	some of the best ice cream we've ever had
	<time> , another server came to take order .	it took over <time> before the server came over to take our beverage order .

Figure 12. Source sentence and the generated sentence using the Baseline VAE model and the ES-NLM

were not only more natural, but also emotionally enriched to include more superlatives and select the more emotionally rich and lexically diverse snippets of sentences via the latent edit vector. This propensity is seen as desired throughout our sentences, especially when compared to the baseline VAE that tends to produce generic sentences that have snippets occurring in many reviews of restaurant-goers of different emotional states. Consequently, the baseline VAE model outputs grammatically sound sentences, but such sentences tend to lack diversity and discernible emotion.

However, the problem of generality was not completely solved by our model. When we evaluate the output sentences of our ES-NLM model in Figure 13, several generated outputs are slight variants of the same words and phrasing, making it difficult for our model to generate a sentence with an emotional predilection. Although we were able to success in several instances, in general we found it difficult to generate high quality outputs with the opposite emotional specification as the input and our model largely refused to provide a positively scored candidate. This is likely a result of the model's failure to consistently identify latent variables that can map a positively categorized source sentence into negative counterparts due to the large Jaccard distance between such sentences and our limitation of the search space to a relatively narrow neighborhood. Consequently, such conversion can generally not be edited in one try, requiring multiple edits to traverse through the latent space to find an appropriate counterpart.

Despite this limitation, the outputs of the ES-NLM were rated most likely to appear to in the Yelp Review as either positive or negative sentiment user reviews. Consequently, the ES-NLM model has generated sentences that come closer to achieving a human-like quality compared to previous models that build sentences from scratch or focus strictly on content quality and not on emotions.

Source Sentence	Generated Sentence
"we went there for the <ordinal> time recently and was amazed."	"we went there for the <ordinal> time recently and was pleasantly surprised ."
"'i give this place <cardinal> stars for the food alone ."	"in all honesty , i really wish i could give this place <cardinal> stars !"
"Based on our experience, I wouldn' want anyone to go <org>"	"based on our experience , i do n't recommend <org> to anyone ."]
"The drinks were tasty and it was also strong"	"the drinks were pretty good and strong"

Figure 13. The new output sentences generated using ES-NLM. All these sentences. Based on the survey, three out of four of these sentences were rated the most positive or negative sentences to appear in a Yelp review, and one of the option was close second.

8. Conclusion & Future Work

Text generation is undoubtedly a difficult task, fully capturing and reproducing the complexity of the human language is still a work in progress. We introduce ES-NLM, a variant of the prototype-then-edit model, that learns to generate an edited sentence from a prototype sentence conditioned on some emotional specification and the original semantic similarity between the source and target sentences. Through this research. The ES-NLM learns to construct an edited sentence that comes closest to a user-specified emotion, in this case simply positive or negative emotion.

We believe that ES-NLM can be generalized to a wider range of inputs and can be more emotionally expressive once we train our model on a more diverse and larger dataset. Therefore, for future work we plan to explore these possibilities and better measure current model's performance with more time and computational resources on hand. Moving forward, we would also like to research ways to more efficiently and effectively construct edit vectors that intuitively correspond to categories of human emotion. Accomplishing this would vastly expand the emotional capabilities of current IPAs and chatbots, paving the way for more personal and natural human-machine conversations.

9. Acknowledgements

We would like to thank the CS 224U teaching team, particularly Akhila Yerukola, for their support with our project. We used CodaLab (5) compute resources to execute our code.

10. Authorship Statement

All members contributed to the research and writeup. ST executed ES-NLM experiments on CodaLab, JJ focused on sentiment analysis and baseline experiments, and NM focused on analyzing the baseline and ES-NLM data to obtain results. All members contributed proportionally to written and video materials.

All code and materials available at:
https://worksheets.codalab.org/worksheets/0x5c1e998c3d09419a895fd5f006ced331

References

[1] A. Agrawal, A. An, and M. Papagelis. Learning emotion-enriched word representations. *International Conference on Computational Linguistics*, 27:950–961, 2018.

[2] T. Antonova. Vae-text-generation. https://github.com/Toni-Antonova/VAE-Text-Generation/blob/master, 2017.

[3] S. R. Bowman and L. Vilnis. Generating sentences from a continuous space. *arXiv:1511.06349*, 2015.

[4] S. Ghosh, M. Chollet, E. Laksana, L.-P. Morency, and S. Scherer. Affect-lm: A neural language model for customizable affective text generation. *arXiv*, 2017.

[5] K. Guu, T. B. Hashimoto, Y. Oren, and P. Liang. Codalab: Generating sentences by editing prototypes (tacl/acl 2018). https://worksheets.codalab.org/worksheets/0xa915ba2f8b664ddf8537c83bde80cc8c/.

[6] K. Guu, T. B. Hashimoto, Y. Oren, and P. Liang. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450, 2018.

[7] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013.

[8] E. Loper and S. Bird. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

[9] K. Papineni et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144*, 2016.

[10] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. *EMNLP*, 2014.

[11] G. A. . M. L. Products. Evaluating models, 2019.

[12] O. Vinyals and Q. V. Le. A neural conversational model. *International Conference on Machine Learning*, 37, 2015.

[13] Y. Wu et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144*, 2016.

[14] J. Xu and G. Durrett. Spherical latent spaces for stable variational autoencoders. *arXiv:1808.10805*, 2018.

[15] Yelp. Yelp dataset challenge, round 8. https://www.yelp.com/dataset_challenge, 2017.

[16] H. Zhou, M. Huang, T. Zhang, X. Zhu, and B. Liu. Emotional chatting machine: Emotional conversation generation with internal and external memory. *Association for the Advancement of Artificial Intelligence*, 2018.