Formality Neural Machine Translation

Katie Cason, Angela Gao, Julie Lai
W266 Natural Language Processing
UC Berkeley, School of Information
{katiecason, agao729, julielai}@berkeley.edu

Abstract

In a time where interaction is becoming more and more virtual, it is important to make sure the formality and tone of the text is appropriate. Our project explores Neural Machine Translation models to translate informal sentences into formal sentences. Our results show that by using a Long Short Term Memory Sequence-to-Sequence Model with Embeddings, Masking, and two Time Distributed Dense Layers with Dropout, we were best able to translate a sentence while maintaining meaning and fluency.

1 Introduction

Machine Translation models already exist on a spectrum of Neural Machine Translation to Phrase-Based Machine Translation to Rule-Based Machine Translation models to translate different languages. Furthermore, there have already been several Neural Machine Translation models, on both a word and character level, that translate different languages. However, notions of style transfer and specifically formality translation are much less common. While languages generally have common direct translations from one to another, formality does not have direct relationships between informal to formal language. Most words on their own are neutral in tone, and the proper balance of tone and formality can be highly reliant on context. In turn, this can result in very different interpretations. Particularly for the English language, formal language tends to be more circuitous and contain extra words or phrases that do not add more in context or meaning. For example, phrases like "To whom it may concern" and "Sincerely" may be added to make a neutral sentence more formal.

Our project aims to use and compare Neural Machine Translation models for translation of informal English sentences to formal English sentences. Initially, we imagined this functionality to be useful for new English speakers who are trying to navigate the way native English speakers speak informally. However, this model could also be useful for any individual who wants to make their text more formal, such as for emails or for academic papers.

Our approach for formality translation was to use a Long Short-Term Memory (LSTM) Sequence-to-Sequence (Seq2Seq) Neural Machine Translation (NMT) model and compare different modifications of this model.

2 Data

We initially aimed to use MultiUN Data at

https://opus.nlpl.eu/MultiUN.php

and OpenSubtitles2016 at

https://opus.nlpl.eu/OpenSubtitles-v2016.php

for formal and informal samples, respectively, as followed by Niu et al. (2017). However, because these datasets are not direct translations of each other, we would have to create our own word alignment for informal and formal words.

We opted for the Grammarly's Yahoo Answers Formality Corpus (GYAFC) built from the Yahoo Answers corpus: L6 - Yahoo! Answers Comprehensive Questions and Answers version 1.0 (Rao and Tetreault, 2018).

https://github.com/raosudha89/GYAFC-corpus

This corpus gives us a total of 104560 samples of direct informal to formal translations. The GYAFC corpus is divided into two subjects: (1)

Entertainment and Music, and (2) Family and Relationships. These categories were chosen from the Yahoo corpus as they contained the highest prevalence of informal language. Rao and Tetreault (2018), authors of Dear Sir or Madam, May I Introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer and creators of the GYAFC Dataset, had a team rewrite one hundred thousand informal sentences from questions and responses posted publicly on Yahoo! Answers into formal language following predefined criteria. Their corpus includes various model translation outputs and human rewrites for a team to score against the original informal input.

3 Background

In the realm of formality style transfer, Rao and Tetreault (2018) used both Neural Machine Translation (NMT) and Rule-based or Phrase-based models (PBMT). Models tended to make more extensive rewrites compared to the human translations which tended to change the original sentence meaning as well. Though PBMT made smaller adjustments, the formality increase was also less. Regarding the trade-off between original meaning preservation and improved formality, we decided to focus more on increased formality adjustments.

For alternative applications, Ge et al. (2018) used an RNN based encoder-decoder seq2seq model with multiple seq2seq rounds in order to improve a sentence's fluency. They tried different fluency boosting strategies (back-boost, self-boost, and dual boost) for Neural Grammatical Error Correction.

A similar study in machine translation formality has been done by Niu et al. (2017) using cosine similarity to define lexical formality and use a 4-gram language model for their formality-sensitive machine translation. Their Formality-Sensitive Machine Translation (FSMT) takes in text in the source language and a desired formality level to output a language translation with the appropriate formality.

4 Methods

4.1 Data Processing

To prepare our data for our NMT models, we first modified our datasets to follow the following format:

```
<informal input> \t
<formal translation> \t
<formal translation> \n
```

We then tokenized each sample to be a sequence of integers on both a character and word level.

4.2 Baseline Model

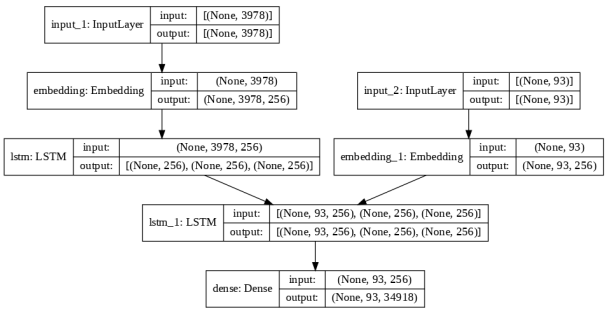

Figure 1: Structure of Baseline Model for both Character and Word level

The architecture of an LSTM is the most appropriate technique for the problem we are trying to solve largely due to the LSTM's capability to learn long-term dependencies. Informal to formal sentence pairs commonly result in large steps between aligned words or phrases, especially with the addition of "filler" words.

The state-of-the-art for Seq2Seq Neural Machine Translation provided by Keras, specifically for translating English to French, uses a basic Long Short Term Memory (LSTM) setup and tokenizes sequences at the character level. This simplistic setup is very effective for translating English sentences to French sentences, but not optimal for our purposes. Because our input data are represented by a sequence of integers rather than one-hot encodings, we added Embedding layers and implemented sparse representation (Chen et al., 2016). This allowed us to increase memory efficiency. Furthermore, because our input data was padded, we implemented masking which allows the model to skip over missing timesteps and increases model accuracy. This modified model will serve as our baseline.

We provide two iterations of this baseline model: first, maintaining the character-level tokenization and second, implementing word-level tokenization. For all subsequent iterations, we maintain word-level tokenization. The decision to transition from character to word level tokenization was made as meaning is better retained within a word than within a character.

4.3 Baseline Model + Time Distributed Dense Layers and Dropout

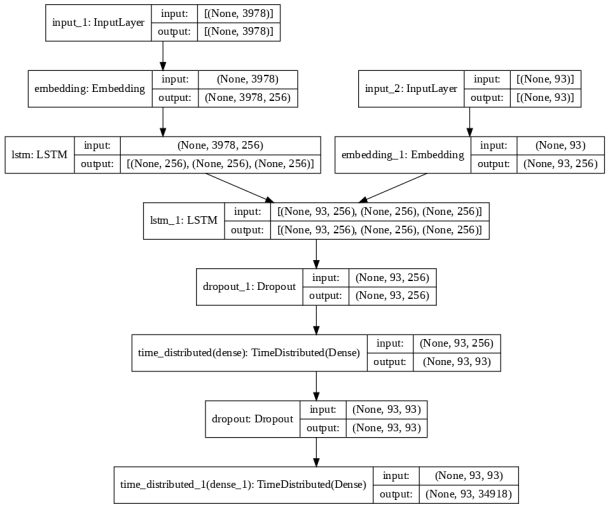

Figure 2: Structure of Baseline Model + two Dense and Dropout Layers

The last iteration of our model changes our final Dense layer to include two Time-Distributed Dense layers and two Dropout layers. Using two Dense layers allows the model to learn more complex features and using Dropout layers serves to keep the model from over-fitting.

5 Results

Noted in Table 1, our baseline character-level LSTM Seq2Seq model achieved a loss of 0.2780, an accuracy of 0.6344, and a validation accuracy of 0.6372. As expected, our baseline word-level LSTM Seq2Seq was a more successful model that better decoded sequences. After ten epochs, this model achieved a loss of 0.3501, an accuracy of 0.7163, and a validation accuracy of 0.6498.

The last iteration with two Dense and Dropout layers trained much slower and did not have as high of an accuracy. However, the decoded sequences are more human-readable. After ten epochs, this iteration achieved a loss of 0.5169, an accuracy of 0.6528, and a validation accuracy of 0.6528 on our test data.

To get a better sense of the models, we compare one input sentence (informal), the human-translated sentence (formal), and the model-translated sentence (formal).

Input: What the guy looks like would not be a big deal, but his personality would matter.
Human-translated: Yes. He was unattractive, as well.
Baseline (character) Model-translated: hat something that you are a good states and the song is a good person who is not a good states.
Baseline (word) Model-translated: the man who is a child, he is always like a good looking for the situation to do it.
Baseline (word) + 2 Dense + 2 Dropout Model-translated: the man looks like a nice person, but she is not interested in him.

5.1 Text Formality Classifier + LIME

Our evaluation metric will be using the Text Formality Classifier built off the GYAFC corpus that examines how different machine learning models perform in classifying a text as formal or informal (Lu and Wang, 2020). We will be using their Multinomial Naive Bayes, Logistic Regression, and LSTM with GloVe classifications. We applied all of the formality text classifiers on our GYAFC formal and informal texts (Table 2) as an indicator of classifier accuracies. An effective Formality Classifier model will score a high percentage in informal classifications for input sentences (informal) and a low percentage for human-translated sentences (formal). We proceeded to use these classifiers on our test set to evaluate how well our NMT models perform on sentences they have not seen before. Each of the three formality classifiers were applied to the test set model outputs from our various translation models (Table 3). A good NMT model will score a low percentage of informal classifications from the Formality Classifier, equivalently high percentage of formal sentences.

To help interpret our various models and their interpretations, we use Local Interpretable Model-agnostic Explanations (LIME) to help explain how notions of formality are represented in our corpus (Ribeiro et al., 2016). LIME allows us to visualize which words have the strongest weight in indicating formality. We chose to visualize how the Multinomial Naive Bayes Text Formality Classifier classifies the different translated outputs on the same input sentence in the section above. We note that ”guy” and ”big” are strong indicators for informality and ”unattractive”, ”situation”, and ”interested” are strong indicators for formality (see Appendix A)

Model	Loss	Accuracy	Validation Accuracy
Baseline (character)	0.2780	0.6344	0.6372
Baseline (word)	0.3501	0.7163	0.6498
Baseline (word) + 2 Dense + 2 Dropout	0.5169	0.6528	0.6528

Table 1: Loss, accuracy, and validation accuracy for the three iterations of the Baseline LSTM Seq2Seq Model. 10 epochs were run on each.

	Multinomial Naive Bayes	Logistic Regression	LSTM w/ GloVe
Informal	77.78%	76.26%	63.31%
Formal	73.74%	77.64%	43.71%

Table 2: Test formality classifier accuracy on GYAFC corpus informal and formal sentences

% Informality on Test Set			
	Multinomial Naive Bayes	**Logistic Regression**	**LSTM w/ GloVe**
Input	74%	70%	63%
Human-translated	31%	41%	65%
Baseline (character) Model-translated	0%	0%	9%
Baseline (word) Model-translated	10%	8%	35%
Baseline (word) + 2 Dense + 2 Dropout Model-translated	7%	7%	23%

Table 3: Text formality classifier evaluating % informality on test set inputs and model outputs.

6 Conclusion

In comparing the models, a word-level LSTM Seq2Seq Model performs better than a character-level LSTM Seq2Seq Model. This makes sense because meaning is better retained in words than in characters. Of the two word-level models, while the Baseline Model has a better accuracy than the Baseline Model + two Dense and Dropout layers, the Baseline Model + two Dense and Dropout layers has a better validation accuracy and creates more sensible translations. Furthermore, the Text Formality Classifier that we use to evaluate our models shows that the translations from Baseline Model + two Dense and Dropout layers score the most formally.

7 Discussion

Due to memory, GPU, and time limitations, we were not able to run as many models as we'd like to run. Included in our files but not run was our model including Attention. We believed that adding an Attention layer would improve our scores because the layer takes into account context (Bahdanau et al., 2016), however we were not able to both tune and run this model due to the aforementioned constraints. We used Google Colaboratory Pro to run our models, as it provided us with a 24 hour runtime limit, high-memory VMs, and access to GPUs. However, a model with 10 epochs took approximately 30 hours to run and Google Colaboratory Pro would disconnect several times within the runtime limit. Given a more powerful machine and more time, we would aim to add Attention, fine tune our models, and train our models for longer.

Acknowledgments

We would like to thank the W266: Natural Language Processing with Deep Learning instructors. In particular, we'd like to thank Peter Grabowski and Joachim Rahmfield for their guidance throughout our entire project. Their help was pivotal to the momentum of our project.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate.

Yunchuan Chen, Lili Mou, Yan Xu, Ge Li, and Zhi Jin. 2016. Compressing neural language models by sparse word representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 226–235, Berlin, Germany. Association for Computational Linguistics.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Fluency boost learning and inference for neural grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1055–1065, Melbourne, Australia. Association for Computational Linguistics.

Xiaoyu Lu and Yonglin Wang. 2020. Capturing text formality of online answers: Analyzing gyafc with machine learning algorithms. `https://github.com/YonglinWang-Brandeis/text-formality-classifier`.

Xing Niu, Marianna Martindale, and Marine Carpuat. 2017. A study of style in machine translation: Controlling the formality of machine translation output. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2814–2819, Copenhagen, Denmark. Association for Computational Linguistics.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144.

A Appendix

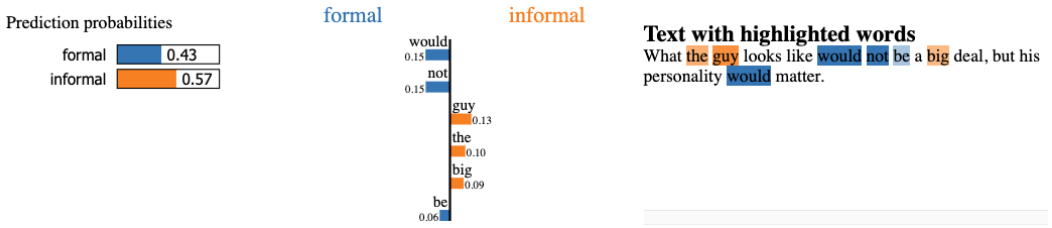

Figure 3: LIME Visualization for Multinomial Naive Bayes Text Formality Classifier on Input (informal) Sentence

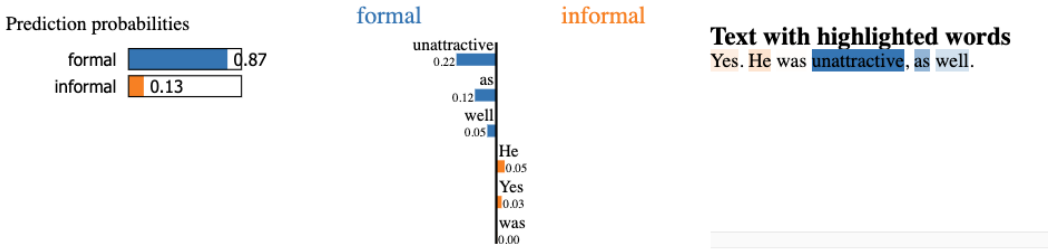

Figure 4: LIME Visualization for Multinomial Naive Bayes Text Formality Classifier on Human-Translated Sentence

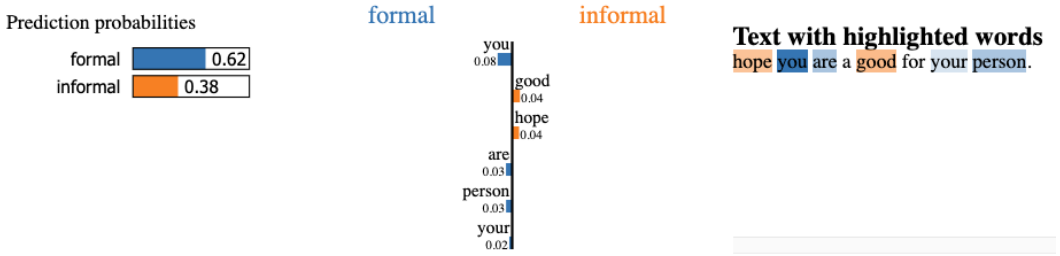

Figure 5: LIME Visualization for Multinomial Naive Bayes Text Formality Classifier on Baseline (character) Model-Translated Sentence

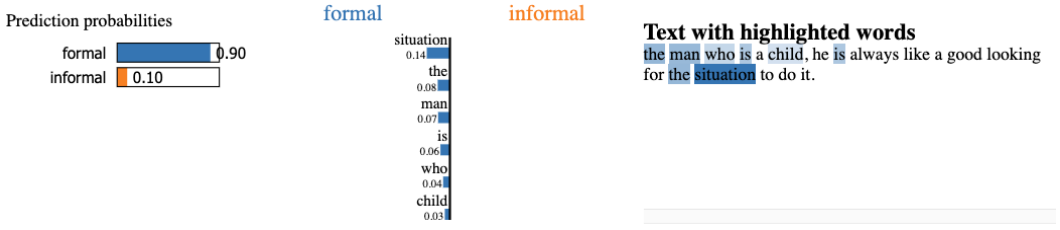

Figure 6: LIME Visualization for Multinomial Naive Bayes Text Formality Classifier on Baseline (word) Model-Translated Sentence

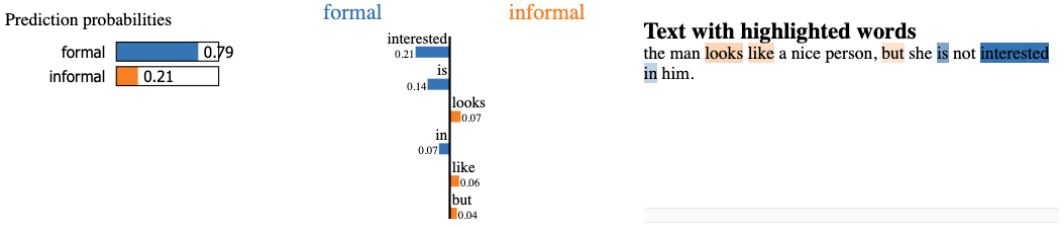

Figure 7: LIME Visualization for Multinomial Naive Bayes Text Formality Classifier on Baseline (word) + 2 Dense + 2 Dropout Model-Translated Sentence