FORMALITY TRANSFER EVALUATION

A PREPRINT

Ankitkumar Joshi
Department of Information Science
University of Pittsburgh
Pittsburgh, PA
anj88@pitt.edu

Vaishnavi Deshapnde
Department of Information Science
University of Pittsburgh
Pittsburgh,PA
vad28@pitt.edu

Yue Dai
Department of Computer Science
University of Pittsburgh
Pittsburgh,PA
yud42@pitt.edu

April 27, 2019

1 Introduction

As the most fundamental way of knowledge representation and communication, languages not only carry the semantic expressions in one restricted style. In fact, there would be multiple ways of presenting similar semantic contents within a language, that is, the same content be represented in multiple different styles in terms of formality and/or sentiment. Expression in proper style is important in the daily life, such as a formal email. Considering the significance to express the content in right way, the text style detection and transferring problem is always an interesting part of natural language processing areas, and gives potential in automatically correction as well as translation.

With the advancement in machine learning, there are growing researches on the text style transferring tasks. The task is of potential use in writing assistant tools such as style formatting and correction, as well as multi-language translation. Though there are many techniques on the automatic style transferring, there is still no consensus formalization for this topic (Tikhonov, et al., 2018). In the recent works, people came up with different rule based as well as neural machine translator based approaches to solve the question. And the standard method to evaluate the output is only compared with gold standard reference from human annotators.

However, there is still no evaluation method clear and comprehensive enough to recognize a good translation output. Therefore, in this project we aimed to rank and evaluate the approaches for the automatic text style transferring problem specific in formality transferring, by analyzing the output in terms of formality, semantic preservation as well as fluency. We explored formality, and ranked the semantic preservation and fluency based on specific matrices by conducting following works:

- **Formality identification**: We identified the formality based on formality scores get from language model and Neural Network Classifier methods.
- **Semantic and fluency check**: We used BLEU and GLEU scores for semantic and fluency check for the output.
- **Proposal**: Based on our analysis on current models, we gave some thought on possible solutions for the text style transfer problems.

2 Background

2.1 GYAFC Dataset

Grammarly's Yahoo Answers Formality Corpus (GYAFC) is the largest dataset for any style containing a total of 110K informal / formal sentence pairs. Yahoo Answers, a question answering forum, contains a large number of informal sentences and allows redistribution of data. The authors of the dataset randomly sample a subset of 53,000 informal sentences each from the Entertainment & Music (E&M) and Family & Relationships (F&R) categories and collect one formal rewrite per sentence using Amazon Mechanical Turk. They collect multiple references for the tune/test set and ensure that they are different rewrites. Below is a prototypical example for E&M and F&R informal and their respective formal human references taken from tune split.

Entertainment & Music

Informal	Any movie that has vampires I like!
Formal (ref0)	I like any movie that has vampires in it.
Formal (ref1)	I enjoy any film that has vampires in it!
Formal (ref2)	I enjoy any film that has vampires.
Formal (ref3)	I like any movie with vampires!

Family & Relationships

Informal	she sees you more as a friend, nothing more
Formal (ref0)	She views you more in a friendship role, and only to that extent.
Formal (ref1)	She sees you as nothing more than a friend.
Formal (ref2)	She thinks of you as no more than a friend.
Formal (ref3)	She sees you more as a friend and nothing more.

2.2 Text Style Transfer

The aim of style transfer is to transfer the text from source style to the target style. The supervised text style transfer includes aligned parallel sources which can be seen as a special case of Machine Translation. Given the input sequence X consisting of words sampled from a style A distribution, find out the corresponding target sequence Y sampled from style B distribution such that it preserves the content of X and is also consistent with the language model of A and B. One example of these paired sequences is as follows.

- IM GLAD THEY PASSED THE TOILETS (Informal)
- I am just glad they didn't show us the toilets. (formal)

The style transfer problem can be divided into two categories: one is dealing with parallel data, which has (X, Y) pairs in parallel such that X is in style A, Y in target style B, and X and Y express the same content. The other category is style transfer using un-parallel data sources. The problem is then defined as given a set of sequences X in style A and a set of Y in style B but the content of X and Y are not necessarily aligned. For instance, a corpus of scientific paper and a corpus of internet blogs. They are indeed different in style but there is no word sequence to sequence relationship.

In this study, we will focus on style transfer with parallel data resources. We evaluate performance of the state-of-the-art MT approaches including rule-based Statistical Machine Translation (SMT), Phrase-based SMT (PBMT), Neural Machine Translation (NMT) for the style transfer problem. Specifically, we evaluate rule-based, PBMT, NMT-baseline, NMT-copy, NMT-combined models. And we will go over them briefly in the following subsection.

2.3 Style transfer models

In this part we give a brief introduction about the translation models implemented on the text style transfer tasks we want to evaluate on. Includes Rule-based MT, PBMT, NMT-baseline, NMT-copy, NMT-combined (Sudha, et al., 2018).

2.3.1 Rule Based Machine Translation

As the oldest approach of machine translation, the rule based machine translator (Nirenburg, Sergei, 1989) restricted follows the Vauquois triangle (B. Vauquois, 1968) following 3 steps: 1) Firstly, do semantic and syntactic analysis based on a rule data based on source sequence X. During this step the translator goes higher in Vauquois triangle into some inter-representations (for example, semantic representations) above languages. 2) When the analysis finished, follow a rule set to transfer the inter-representations into lower representations in target domain. 3) In the end, generate the target text sequence Y based on rules from higher inter-representation to plain texts.

2.3.2 Phrase based Machine Translation

The Phrase based Machine Translator (Koehn P, et al. 2003) are most commonly used currently in the translation problems. Different from rule based ones analyzing and abstract the input sequence X into higher inter-representation format, it uses statistical density distributions like language model in following way: Firstly, it learns and maintains a phrase based mapping (one-to-many) from source domain to target domain. 2) Secondary, it creates a set of candidate sentences on the target style. 3) Lastly, based on the language model trained on target domain, it picks the one candidate with highest probability.

2.3.3 Neural Machine Translation

As the most current techniques to exploit the advancements of neural network models, the Neural Machine Translations (Cho et al., 2014) have shown great potential in the machine translation area, therefore, also implemented in style transferring tasks. The encoder-decoder model formalized input sequence X into higher level hidden state representations by encoder and then recovered into sequences Y in target style. We evaluated following NMT models (Rao et all., 2018) based on GYAFC corpus.

NMT baseline: The baseline model of NMT is a bidirectional LSTM encoder-decoder model with attention (Bahdanau et al., 2014). The NMT model (along with rest) are trained on the rule-based output in GYAFC corpus.

NMT copy: The copy enriched NMT (Jhamtani et al., 2017) is based on the baseline model above, except it involves a pointer model, which takes the attention weights from attention model of baseline encoder-decoder model and copies over features directly for the transfer.

NMT combined: The method increases the number of data samples to improve training performance by using PBMT method back-translation (Sennrich et al., 2016c) to generate more data, then duplicate the previous rule-based data to up weight the original data.

3 Approach

In this section we show the methods we used to evaluate the output from models introduced above. We compared the models in following aspects:

- **Formality**: To evaluate and analyze the formality automatically, we used a statistical language model approach and a neural network approach to quantify formality.
- **Content Preservation**: Describes how well the semantic content of output matched with the target ones.
- **Fluency**: One criterion to check out whether the generated texts are legal English, i.e. Grammatical correctness. To rank the model in terms of content preservation and fluency, we use BLEU and GLEU scores.

3.1 Statistical Model

Following are the word frequency distribution for formal and informal language models:

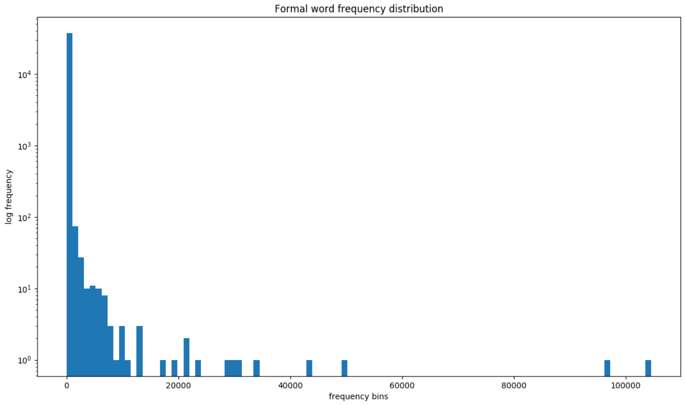


Figure 1: Formal word frequency distribution

We model two different distributions: unigram language model on tokenized words and bigram model on Part-Of-Speech (POS) tags for the words obtained from Stanford Core NLP. Note that we have a separate model for formal and informal corpus. The intuition here is that the unigram language model can tell apart cases of particular words that are likely to

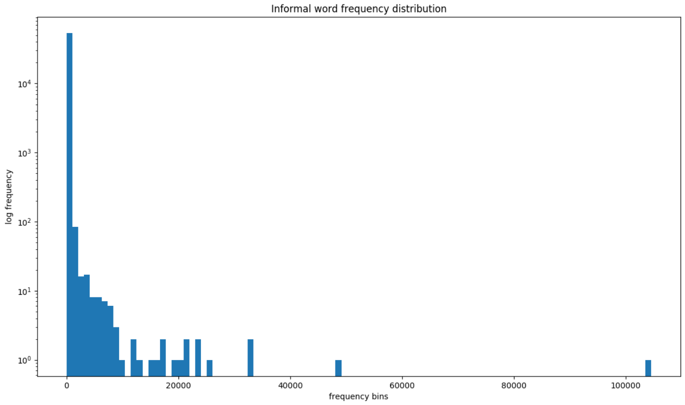


Figure 2: Informal word frequency distribution

appear in informal corpus compared to formal corpus in expectation (cases like 'lol', 'soo'). The bigram model on POS tags will capture grammatical inconsistencies if present. Following equation explain the unigram model. We define an aggregate informality score below as a single number to indicate informality present in the sentence - less is better for informal to formal style transfer task.

$P_{for}^{uni}(w) = \frac{Count_{for}(w)}{Count_{for}(w)+Count_{inf}(w)}$

$P_{inf}^{uni}(w) = \frac{Count_{inf}(w)}{Count_{for}(w)+Count_{inf}(w)}$

$P_{for}^{uni}(S) = \prod_{w_i \epsilon S} P_{for}^{uni}(w)$

$P_{inf}^{uni}(S) = \prod_{w_i \epsilon S} P_{inf}^{uni}(w)$

$P_{for}^{bi}(tag_{w_i}|tag_{w_{i-1}}) = \frac{Count_{for}(tag_{w_i}|tag_{w_{i-1}})}{Count_{for}(tag_{w_i}|tag_{w_{i-1}})+Count_{inf}(tag_{w_i}|tag_{w_{i-1}})}$

$P_{inf}^{bi}(tag_{w_i}|tag_{w_{i-1}}) = \frac{Count_{inf}(tag_{w_i}|tag_{w_{i-1}})}{Count_{for}(tag_{w_i}|tag_{w_{i-1}})+Count_{inf}(tag_{w_i}|tag_{w_{i-1}})}$

$P_{for}^{bi}(S) = \prod_{w_i \epsilon S} P_{for}^{bi}(tag_{w_i}|tag_{w_{i-1}})$

$P_{inf}^{bi}(S) = \prod_{w_i \epsilon S} P_{inf}^{bi}(tag_{w_i}|tag_{w_{i-1}})$

$$P_{inf}^{AGG} = P_{inf}^{bi}(S) * P_{inf}^{uni}(S) \tag{1}$$

In the above equation 'for' stands for formal; 'inf' for informal; 'bi' for bigrams of POS tags; 'uni' for unigram model; 'w' for a token in the tokenized sentence 'S'.

3.2 Neural Network based classifier

The text classification can also be handled by Neural Network classifiers.

3.2.1 CNN Classifier

A CNN classifier for text classification typically uses 1-dimensional convolution window with specified number of filters in each window kernel. The inference process of a CNN classifier can be seen in Figure 3. In our case, we used two 1D convolution layers combined with max-pooling, with pre-trained embedding, to get a categorical prediction on the formality for each text sequences with a softmax output. The texts are firstly preprocessed by tokenizer into one-hot representation in vocabulary size, then shuffled and split into train and test set. We used GloVe embeddings pre-trained on twitter (glove.twitter.27B) to convert the original texts into 100 dimension embedded vector sequences. By padding the sequence into fixed max-length as 100, we get the input data. And with formality tag ([1,0] as informal, [0,1] as formal) based on files, we trained the following CNN classifier. The input texts in one-hot vector sequences

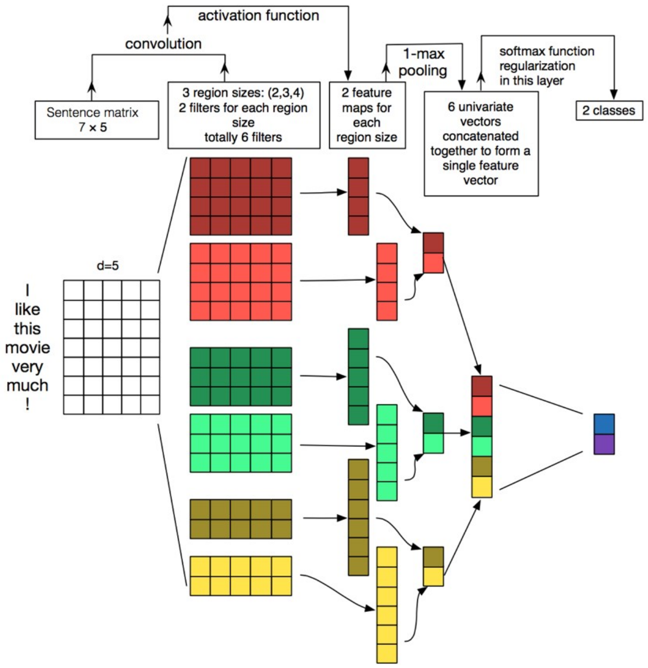
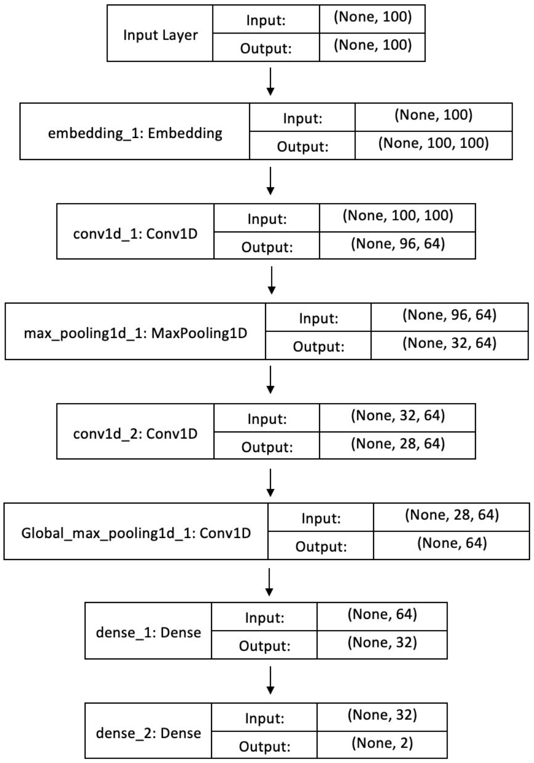

activation function
convolution
Sentence matrix 7 × 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
1-max pooling
6 univariate vectors concatenated together to form a single feature vector
softmax function regularization in this layer
2 classes
d=5
I
like
this
movie
very
much
!

Figure 3: CNN classifier for NLP tasks



Layer		
Input Layer	Input:	(None, 100)
	Output:	(None, 100)
embedding_1: Embedding	Input:	(None, 100)
	Output:	(None, 100, 100)
conv1d_1: Conv1D	Input:	(None, 100, 100)
	Output:	(None, 96, 64)
max_pooling1d_1: MaxPooling1D	Input:	(None, 96, 64)
	Output:	(None, 32, 64)
conv1d_2: Conv1D	Input:	(None, 32, 64)
	Output:	(None, 28, 64)
Global_max_pooling1d_1: Conv1D	Input:	(None, 28, 64)
	Output:	(None, 64)
dense_1: Dense	Input:	(None, 64)
	Output:	(None, 32)
dense_2: Dense	Input:	(None, 32)
	Output:	(None, 2)

Figure 4: CNN classifier Network

format will be padded to 100 maximum length, then fed into embedding layer where each token will be transferred into a 100-dimension embedding vector. Then the sequence will go through two 1d convolution layers; then through global max pooling into 64-dimension feature map for each sequence. Then through 2 forward dense layers, the final output will be a SoftMax vector in Array([Probability to be informal, Probability to formal]) format.

3.2.2 BiLSTM Classifier

Another choice for the neural network classifier is RNN classifier. In here we used a BiLSTM classifier which consist of two LSTM layers. One layer would take the input sequence in forward way in time stamp and another would take the input in a backward way. The final output of the BiLSTM layer would be a concatenation of output from these two LSTM layers. By doing this we could keep more flexible and comprehensive information within the input sequence. The structure can be seen in figure 5. In our case we used a Bidirectional LSTM layer with 50 units each direction, the network details is shown in figure 6. In next we will show how our experiment results.

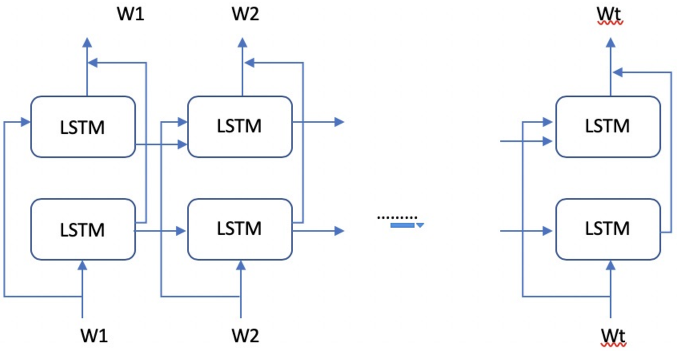
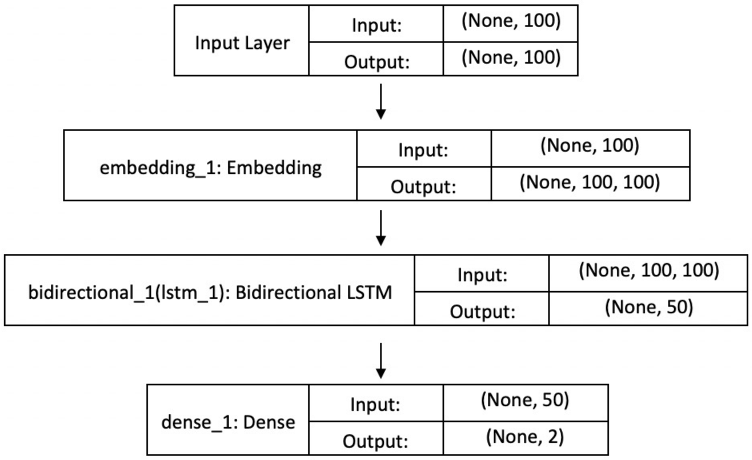


Figure 5: BiLSTM structure



Input Layer	Input:	(None, 100)
	Output:	(None, 100)

embedding_1: Embedding	Input:	(None, 100)
	Output:	(None, 100, 100)

bidirectional_1(lstm_1): Bidirectional LSTM	Input:	(None, 100, 100)
	Output:	(None, 50)

dense_1: Dense	Input:	(None, 50)
	Output:	(None, 2)

Figure 6: BiLSTM classifier Network

4 Experimental Results and Discussion

In this section we discuss the experimental results we obtained.

4.1 Fluency and content preservation

BLEU (Bilingual Evaluation understudy) is an algorithm for evaluating the quality of text which has been machine translated from one natural language to another. For us, it is converting from informal text to formal text.It is a metric which measures the faithfulness by counting the matches, and fluency by implicitly using the reference n-grams as a language model. BLEU's output ranges between 0 and 1. This value indicates how similar the candidate text is to the reference text, with the values closer to 1 representing more similar texts. Natural language toolkit provides packages which can be imported in order to calculate the scores.
Limitations of BLEU are it doesn't consider different types of errors like (insertions,synonyms,paraphrase,stems).And the metric doesn't perform well for single sentences.

$$P = \frac{m}{w_t} \tag{2}$$

where m is the number of words in the candidate that are found in the reference and w_t is the total number of words in the candidate file and P is the precision.
GLEU(Google BLEU) is inspired by BLEU and it also helps to check the semantic preservation and fluency of the translation. Natural Language toolkit provides packages which can be imported in order to calculate the scores.It performs better than BLEU for our dataset.

We have four reference files which serve as gold standard and we compared them against five of our model outputs to calculate the BLEU and GLEU scores. These files are provided with the dataset.

Table 1 shows the Average BLEU and Average GLEU scores for the Entertainment and Music dataset. It shows the model outputs compared against the formal reference files (ref0, ref1, ref2, ref3). As we can see in the table 1, the Average BLEU scores obtained indicate that formal.nmt_combined performed well compared to other model output files. Model outputs formal_baseline and formal.nmt_copy when compared against the references gave us very low scores and it is evident from the table.And formal.rule_based gave intermediate results compared to others.
GLEU scores also gives the results similar to that of BLEU for our model's outputs as shown in Table 1.

Table 2 shows the Average BLEU and Average GLEU scores for the Family_Relationship dataset. It shows the model_outputs compared against the formal reference files (ref0, ref1, ref2, ref3). It shows the model outputs compared against the formal reference files (ref0, ref1, ref2, ref3).As we can see in the table 2, the Average BLEU scores obtained indicate that formal.nbt_combined performed well compared to other model output files. Model outputs formal_baseline,formal.nmt_copy and formal.rule_based when compared against the references gave us very low scores and it is evident from the table.

Candidate	Average BLEU score	Average GLEU score
Formal.nmt_baseline	0.155299	0.25831025
Formal.nmt_combined	0.1855662	0.2879135
Formal.nmt_copy	0.161993	0.2632405
Formal.pbmt	0.1826947	0.28524725
Formal.rule_based	0.1681862	0.2678305

Table 1: Entertainment and Music

Candidate	Average BLEU score	Average GLEU score
Formal.nmt_baseline	0.19798625	0.297602
Formal.nmt_combined	0.21926425	0.3197305
Formal.nmt_copy	0.19834225	0.29797075
Formal.pbmt	0.21072025	0.31347525
Formal.rube_based	0.2013855	0.301219

Table 2: Family_relationships

4.2 Statistical Model

Dataset	Formal acc	Informal acc
Entertainment Music	0.865	0.843
Family Relationship	0.887	0.811

Table 3: Statistical model accuracy results

We used train and tune set for training both the language models and test it on the test set for each category. Table 3 summarizes results on these. We obtain the unigram, bigram POS and aggregate informality on the model outputs shown in table 4. The results suggests that NMT baseline and NMT copy does the best on the aggregate informality score. Here are some examples from NMT model outputs:

Model	Unigram informality	Bigram informality	Aggergate informality score
Rule based	0.512	0.451	0.283
PBMT	0.189	0.293	0.087
NMT Baseline	0.137	0.1787	**0.0352**
NMT Combined	0.188	0.289	0.0865
NMT Copy	0.127	0.182	**0.0345**

Table 4: Statistical model informality score for each style transfer method

- "If you can answer that , then it is the same for the egg ."
  P(Informal Bigram): 0.012
  P(Informal Unigram): 0.027
  Informal: False

- "I used to play flute but once I started saxophone ."
  P(Informal Bigram): 0.521
  P(Informal Unigram): 0.232
  Informal: True

4.3 Neural Model

For the training of classifiers, we used the data pairs in train and tune in both music and family categories. We randomly shuffle the formal and informal data in the train set separately, then sampling from them to get the training data. The training data is randomly shuffled again and sliced into train and valid data in 8:2 ratio. Then I managed to test on isolated test data in the corpus.As we mentioned in previous section. The text would be converted into embedded format via an pre-trained GloVe embedding. And we did following preprocessing on text: Firstly, tokenize the input text and convert them into one-hot vector index format with maximum as vocabulary size minus one (here I use 40,000 MAX_VOCAB). Then we padded them into fixed sequence length (100 MAX_SEQLEN I used here). After preprocessing we leverage the data into models for training.

The classification report for CNN and BiLSTM classifiers are shown in Table 5. The test accuracy reached about 0.78 for CNN classifier and 0.77 for BiLSTM classifier. One possible problem is that the unknown word might be the same in pre-trained word embedding, which cause the information loss from original data. Another reason would be the size of the corpus, it only in size of 6.7MB.

Here we could have some sample to show the probability of being informal with several example pairs from the corpus in Table 6 and Table 7. In the end we used our classifiers to evaluated outcome from machine translator models we mentioned before to compare how well they performed in terms of formality. To do that we first predicted the formal/informal label of model output, then get the probability of being correct. (i.e. If target style is correct, classify the outputs and get the score by dividing amount classified as formal with total numbers of samples.) The final comparison result are shown in Table 8. As we can see, nmt_copy outperformed rest.

Model	CNN		BiLSTM	
	Precison	Recall	F-1 Score	Support
CNN classifiers on Informal	0.78	0.73	0.75	11152
CNN classifiers on formal	0.78	0.82	0.80	13093
BiLSTM classifiers on Informal	0.75	0.80	0.77	11152
BiLSTM classifiers on formal	0.82	0.77	0.79	13093

Table 5: Final test classification report for Neural Network Classifiers

Informal Sentence	CNN Informal Probability	BiLSTM Informal Probability
he iss wayyyy hottt	0.58586705	0.7024421
I've watched it and it is AWESOME!!!!	0.75285065	0.6789353
Well... Do you talk to that someone much?	0.5852287	0.796499
that page did not give me viroses (i think)	0.34214425	0.73979324
my exams r not over yet	0.9917663	0.9861081

Table 6: Statistical model informality score for each style transfer method

Informal Sentence	CNN Informal Probability	BiLSTM Informal Probability
He is very attractive.	0.00152366	0.009885
I viewed it and I believe it is a quality program.	0.02221492	0.0105941
Do you talk to that person often?	0.14816764	0.21179946
I don't think that page gave me viruses	0.5507869	0.6049505
My exams are not over yet.	0.47981066	0.45664433

Table 7: Statistical model informality score for each style transfer method

Model	CNN		BiLSTM	
	Informal	Formal	Informal	Formal
Rule based	0.40	0.55	0.54	0.37
PBMT	0.24	0.74	0.37	0.55
NMT_baseline	0.4	0.83	0.63	0.68
NMT_Copy	**0.43**	**0.84**	**0.63**	**0.70**
NMT_Combined	0.34	0.75	0.50	0.58

Table 8: Model output correctness in formality based on CNN.

4.4 Overall comparison

Combined the evaluation in 3 aspects we proposed we can get an overall comparison shown in Table 9. As we can see the NMT_combined model has the highest BLEU and GLEU score which is the best, the NMT_copy has the best performance in formality, with lowest language model based informal probabilities and highest neural network based formal probabilities.

Model	BLEU	GLEU	Aggergate informality Score	CNN	BiLSTM
Rule_based	0.185	0.285	0.283	0.55	0.37
PBMT	0.196	0.299	0.087	0.74	0.55
NMT_baseline	0.180	0.269	0.0352	0.83	0.68
NMT_Copy	0.181	0.280	**0.0345**	**0.84**	**0.70**
NMT_Combined	**0.203**	**0.304**	0.0865	0.75	0.58

Table 9: Model Comparison Summary

5 Conclusion

In conclusion, we evaluated different machine transfer models for the task of formality style transfer in three aspect: fluency, semantic preservation and formality. We used BLEU and GLEU score to evaluate fluency and semantic preservation, then we designed our own language model and neural network based classifier for formality evaluation. Generally speaking, neural machine translation family outperformed the rest. The nmt_combined model has the best performance in fluency and semantic preservation and nmt_copy model has the best performance in terms of formality. The formality metrics that we describe in this paper do need to be justified with a more rigorous significance test with respect to a human annotated formality/informality score. Given the timeliness of the project, we couldn't incorporate it in our experiments. Statistical model can localize which words in the sentence contributes to informality. As an extension, it would be interesting to find a substitution for the informal phrases obtained from the statistical model.

References

[1] Tikhonov,A., Yamshchikov, I. P. (2018) What is wrong with style transfer for texts?. In *arXiv preprint arXiv:1808.04365.*

[2] Vauquois, B. (1968). A survey of formal grammars and algorithms for recognition and transformation in machine translation', A survey of formal grammars and algorithms for recognition and transformation in machine translation' In *ifip congress-68, edinburgh, 254-260; reprinted in ch. Bernard Vauquois et la TAO: Vingt-cinq Ans de Traduction Automatique-Analectes, 201-213.*

[3] Nirenburg, Sergei (1989) "Knowledge-Based Machine Translation" *Machine Trandation 4 (1989), 5 - 24. Kluwer Academic Publishers. 4 (1): 5–24. JSTOR 10.2307/40008396.*

[4] Koehn, P., Och, F. J., Marcu, D. (2003, May) Statistical phrase-based translation. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1 (pp. 48-54). Association for Computational Linguistics.*

[5] Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. *arXiv preprint arXiv:1409.1259..*

[6] Rao,S., Tetreault, J. (2018). Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. *arXiv preprint arXiv:1803.06535* .

[7] Bahdanau, D., Cho, K., Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473.*

[8] Jhamtani, H., Gangal, V., Hovy, E., Nyberg, E. (2017). Shakespearizing modern language using copy-enriched sequence-to-sequence models. *arXiv preprint arXiv:1707.01161.*

[9] Sennrich, R., Haddow, B., Birch, A. (2015). Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709.*

[10] Zhang,Y., Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820.*

[11] Papineni, K., Roukos, S., Ward, T., Zhu, W. J. (2002, July). BLEU: a method for automatic evaluation of machine translation. *Proceedings of the 40th annual meeting on association for computational linguistics (pp. 311-318). Association for Computational Linguistics.*

[12] Mutton, A., Dras, M., Wan, S., Dale, R. (2007). GLEU: Automatic evaluation of sentence-level fluency. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (pp. 344-351).*

[13] Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations (pp. 55-60).*