Multi Politeness-Domain Neural Machine Translation for Japanese and Korean

Henry Li Xinyuan, Jerry Chen, Ray Lee

Autumn 2021

1 Motivation

Language can work in extremely subtle ways, and sometimes very small changes can drastically alter the undertone of a sentence or an utterance [FHG19] [NRC18]: tiny details such as intonation, conjugation, choice of vocabulary, as well as many factors that would often be overlooked by neural models when extracting meaning [PT16]. While general sequence-to-sequence machine translation is capable of producing results that are very coherent and semantically close to the target output, small details such as formality can often be neglected by the model [RT18] - details that can make or break a career if approached carelessly by a human. As such, we propose systems that aim at adapting existing neural machine translation architectures, so that they would be able to classify sentences according to their formality and reproduce sentences in the correct formality class.

2 Training Data

2.1 Choice of Corpus

Countless corpora of Japanese exist on the Internet, yet the ones that would be suitable for our needs are far and few between. There is typically a strong correlation between formality and context, which is not bad news for us since relying purely on morphology to label formality would have problems of its own. However, we want to avoid introducing into our corpus large chunks of sentences with the same context in the same formality domain, lest any of our models learns to classify contexts rather than formality. Many such examples of bad corpora exist, such as the corpus of Japanese legal documents: in Japanese, all legal documents are written in informal form (contrary to what one might assume); we must be very careful when using such corpora by balancing and mixing them with corpora from other sources and with different formality domains. Examples of good corpora include the subtitle corpus, although the translation quality of some of the sentences in that corpus has been questioned. In general, Japanese is a more context-based language, and it often leaves out parts of the phrase than can be inferred. In most cases, this means omitting the subject. This means that English translation will generally have more context than the Japanese equivalent. Currently, there are no corpora that avoid this issue, as any such corpus would have to be either laboriously picked out or have a heavy bias toward formal sentences. As such, it may be that Japanese to English translation "predicts" a subject, as one is required in English.

2.2 Politeness Labels

We designed our model to be able to handle both translation and formality classification. While extracting a representation for politeness from the automatically extracted features in a neural network pipeline isn't impossible, that is not what we are trying to achieve. Rather, we would train our model under a supervised learning scheme, where each sentence has a corresponding ground truth translation and formality label attached.

One of the earliest roadblocks we faced is the scarcity of such sentence-formality pairs. Such corpora are extremely difficult to find in sufficient quantities that would allow for adequate training of a neural classification model. Human annotation is unfortunately not so accessible for Japanese (in terms of pricing) as some other languages. As such, we devised a number of ways to generated such sentence-formality pairs.

2.2.1 Procedural Generation of Politeness Labels

Fortunately for us, this is a topic that had been studied previously by Feely et al. [FHG19], which in turn was based on the Kyoto Text Analysis Toolkit (KyTea) [NNM11]. In their case, Japanese was the destination language; the task was to generate Japanese sentences that would match the formality levels of the input.

The authors identified verb suffixes and copulas as the keys for indentifying sentence formality, with short-form corresponding to informal sentences and long-form corresponding to formal ones. They published a conversion script which would identify and convert any informal verb suffixes into formal verb suffixes, and vice versa. There is a slight complication to this rule, which is that in Japanese there are situations which constrain the verb to be short-form, ie. when it forms part of a verb phrase that isn't the head of the sentence. Fortunately, Japanese is also a language which observes the SOV order, meaning that the final verb in a sentence is always the head verb and is never constrained to short form. As such, we can first perform sentence tokenisation and then use the long-short form of the final verb/copula in each sentence to determine its formality. There are some exceptions to this SOV structure, especially in the short-form, and especially involving the omission of the verb "desu", (translated as "to be"). Fortunately, in these cases, the final term is an adjective, which in Japanese also conjugate in a way that's distinguishable by politeness.

We thus from their script and make two important modifications. First we adapt their conversion script into a classification script. We also take advantage of the fact that the formality level in a single document should remain the same. This observation can serve multiple purposes:

1 A sanity check for the outputs of our script;

2 Simplify our calculation for classifying sentences that had been segmented into documents.

We performed a sanity check on the legal corpus we mentioned earlier (a corpus entirely made up of informal sentences) and got very positive results: not a single sentence in the corpus of 262000 lines were identified as formal.

Some corpora are already tokenised at sentence level and do not form part of a document, and we were forced to weaken our common-formality assumption to individual corpus entries (which may contain multiple sentences).

One problem that we faced with Japanese-English translation was that Japanese is a language with strong formality and honorifics and English is a language with very weak formality. Thus, we sought to find something similar to Feely et al.'s work, but in Korean. Bravender's open-source application on conjugating Korean verbs proved useful to us.

Bravender's application, koreanverb.app, takes in a Korean verb as an input, and outputs the stem and different conjugations, taking into account tense, formality, and types of statements, such as declarative, imperative, and so on. We believed that this tool would be useful in determining whether the Korean sentences in the corpora we choose to use are formal or informal.

Since Korean is a language that observes SOV order like Japanese, we followed similar steps as we had done previously for analyzing the formality of a given Korean sentence. After tokenizing Korean sentences, we would detect the last verb in the sentence. Then, by using methods from Bravender's application, we would analyze the verb to see whether it was in a formal (or high, in Bravender's code) form or an informal (or low) form.

2.2.2 Using Pre-trained Japanese Language Model for Politeness Labelling

While Japanese is not nearly as high-resource as English (GPT-3 was never specifically trained on Japanese, for example), there are nevertheless some available pre-trained Japanese language models that are available. One of the best recent models that were developed is the Japanese BERT trained at Tohoku University. Similar to the original BERT, this model performs a mask-filling task on Japanese sentences.

The obvious way of making use of a pre-trained language model for politeness labelling would be to fine-tuned it on the new task. However, that would require correctly labelled data, leading us to a chicken-and-egg problem. Alternatively we could use the labels that we generated in section 1.2.1, although then we would be constrained by the quality of our previous scoring function.

A slightly modified approach would be: for each sentence, we identify and mask each verb (along with its suffix conjugation) and each copula one at a time. We feed the masked sentences to the language model. We then score each sentence based on whether the language model filled the masks with long forms or short forms.

2.2.2.1 BERT Fine-tuned on Formality Classification The setup for this experiment is simple: we take a pre-trained Japanese language model and attempt to fine tune it with the sentence-formality label pairs that we generated in section 1.2.1. We first try viewing this problem as a scoring problem, with the model tuned to produce a score between 0 and 1 for each document and with a score greater than .5 indicating that the model considers the document to be formal, and vice versa. We tried two different experimental settings, one where ambiguous sentences (those that didn't contain a formality indicator) are mixed in with target score .5, the other where ambiguous sentences are not included.

The other approach is to regard the problem as a two-class classification task. We do not introduce a third class of ambiguous sentences under this setting, since it is questionable from a linguistic point of view to label these sentences as "formality-free". This is the approach which we experimented on.

The pre-trained model that we used was BertJapanese, developed by Masatoshi Suzuki at the Tohoku NLP lab and available in the transformers package by huggingface. The model converts a character-tokenised sentence into a 768-dimension embedding which we use as the output for our classifier. Given the low dimensionality of our output, we choose the simplest model architecture possible: a linear layer and a log softmax output processing layer.

In our experiments, we were able to achieve a labelling accuracy of .824 for our generated formality label data. We observe similar movements in train and dev accuracy and loss during training, as well as comparable test accuracy compared to train and dev accuracy. Given the size of our test data set (152576), we believe our results are robust and not the result of any fortuitous statistical event.

We compare those results to the ones we achieved from a BERT model based on word tokenisation. The accuracy was comparable to the character-based BERT, at .816, however there was a substantial runtime reduction due to word-level tokenised sequences being shorter than character-level tokenised ones (average length of a Japanese word is around 2 characters, however we would expect -a and did observe - speedup less than 50% due to punctuations, numbers, non-Japanese characters, etc.).

2.2.2.2 Masking Decisive Verb/Copula This setup doesn't even require any fine-tuning, simply deploying existing mask-fill language models and masking the correct words would suffice.

2.2.3 Other Techniques for Politeness Labelling

Some other techniques for politeness labelling of Japanese and other languages had been proposed, and we will discuss them briefly here. Dugan [Dug20] proposed generating politeness labels from the corresponding English translation, an idea we didn't find convincing due to the inherent problems associated with inferring formality from English which has relatively few clear markers for formality, not the mention the problem with noise introduced by otherwise perfect translations with incorrect formality.

3 Model Architecture

A number of possible architectures could be used to integrate the translation task with the formality labelling task. We implement a number of thems, comparing their performance when their tasks are comparable.

3.1 Base translation model without formality labels

We first build a base translation model that doesn't take formality into account. We will use its performance as a baseline and attempt to beat its performance with our formality-aware model. For test cases where either the source or the target language doesn't have strong formality classes (such as Japanese-English or vice versa), we are only interested in the translation quality. For cases where both the source and the target language have strong formality classes (such as Korean-Japanese), we are also interested in formality class correspondance.

The base translation model we choose for our task an autoregressive transformer model [Vas+17], which was selected due to its success in machine translation [Liu+20]. The encoder-decoder construction is significant for us, since the pooled encoder output is going to be the basis for our formality classifier (similar to what we did with the pre-trained Japanese BERT).

For Japanese language pre-processing, we relied on the sentence tokenisation tool by Mecab [Kud05]. For pre-processing English sentences, we used the tokenisation tool by fairseq [Ott+19].

3.2 Base Autoencoder-Reconstructor

This can be seen as a special case of the base translation model, except that the source and the target languages are identical and we measure its ability to reconstruct the original sentence as well as the correct formality label. It follows the same model architecture as the base translation model. During our experiments, we found that the number of transformer encoder and decoder layers had a negative effect on the model's ability to perform the sentence reconstruction task: with the default number of layers (6), the model would emit the highest-frequency word in every position. This is potentially due to the relative simplicity of the task and the model's large size, the combination of which caused a propsentiy towards overfitting.

On our Japanese corpus, we were able to achieve a BLEU score of .648 when benchmarked against the original sentence.

3.3 Adding the Classifier

We decompose the usual encoder-decoder, using the encoder output as the basis for the formality labelling task. We build a separate classification module which takes encoder output as input and produces a formality label prediction. During training time, the loss function is constructed as a weighted sum of the translation loss and the formality label loss.

3.3.1 Autoencoder-Reconstructor plus Classifier

The choice of the hyper-parameter representing the weight of formality label loss in the total loss was initially chosen to be .5. We found that under such a split, the reconstruction performance was barely affected affected (.632 BLEU score, compared to .648 without classification loss). The classification accuracy at .800 is somewhat poor, comparable to what we achieved from the pre-trained encoder .816, but disappointing when considering that the encoder was being fine-tuned under this new setting.

Is the translation task interfering with formality classification? We try to answer that question by turning off the decoder and training the model only on the classification loss. To our surprise, classification accuracy remained roughly the same with the decoder turned off, at .799. There are multiple potential reasons for such an observed effect: it is possible that the transformer encoder (or token embedding sequences in general) do not preserve sentence formality well; it could be that sentence formality is preserved, but not linearly and therefore our classifier was unable to extract that information; finally, the noise level of our procedurally-generated formality labels may have played a role in limiting the potential of any classifier trained on our data.

3.4 Formality Autoencoder

This setting can been seen extending the base autoencoder setting. Here we combine the encoder output with the output of the classifier and feed it back to a decoder which emits a sequence in the source language. We then measure how well the model preserves both the meaning and the formality of the original sentence.

3.5 Formality-Aware Translation

This is the generalised version of the formality autoencoder, where the target language is different from the source language (finally the true translation setting). We compare the results with the base translation model without formality labels.

3.6 Formality-Aware Multi-Channel Autoencoder-Reconstructor

3.7 Formality-Aware Multi-Channel Translation

4 Evaluation

We propose a number of different ways to evaluate the translation quality and the sentence reconstruction quality for our different tasks. We analyse the level of correlation between these different metrics on different tasks.

4.1 Alignment BLEU score

We use the NLTK toolkit [BLK09] to evaluate the BLEU score of our translations or sentence reconstructions. BLEU score is calculated as the proportion of n-gram overlaps between the hypothesis and the reference; due to only having a single sentence per training data point, such a score is not necessarily a good estimate of the model quality. It also doesn't have a good correlation with sentence formality.

4.2 Sentence Embedding Similarity

Comparing the embedding similarity of two sentences is hardly a trivial problem. In order to make our evaluation task tractable, we use a rather trivial construction where we define the embedding of a sentence as the average embedding of all word embeddings in the sentence. We then define sentence similarity as the dot product between the embedding of the two sentences.

4.3 Formality Class accuracy

Our model's understanding of formality is undoubtedly one of the most important aspects of its performance. There are two different angles to formality: the model's classification of input formality, as well as the correctness of the formality of the sentence generated by our model with respect to the original sentence. The former is relatively straightforward to evaluate; the latter requires either running the output sentences on the procedural formality classification script we used to produce our corpora, or to have someone familiar with the Japanese (or Korean, English) language evaluate them manually.

References

[BLK09] Steven Bird, Edward Loper, and Ewan Klein. *Natural Language Processing with Python*. O'Reilley Media Inc., 2009.

[Dug20] Liam Dugan. "Learning Formality from Japanese-English Parallel Corpora". MA thesis. University of Pennsylvania, 2020. URL: http://liamdugan.com/static/thesis-bb3055b500391857a7b237a22a6f18e5.pdf.

[FHG19] Weston Feely, Eva Hasler, and Adrià de Gispert. "Controlling Japanese Honorifics in English-to-Japanese Neural Machine Translation". In: *Proceedings of the 6th Workshop on Asian Translation*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 45–53. URL: https://www.aclweb.org/anthology/D19-5203.

[Kud05] Takumitsu Kudo. "MeCab : Yet Another Part-of-Speech and Morphological Analyzer". In: 2005.

[Liu+20] Xiaodong Liu et al. *Very Deep Transformers for Neural Machine Translation*. 2020. arXiv: 2008.07772 [cs.CL].

[NNM11] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. "Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis". In: *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*. Portland, Oregon, USA., 2011. URL: http://www.phontron.com/paper/neubig11aclshort.pdf.

[NRC18] Xing Niu, Sudha Rao, and Marine Carpuat. "Multi-Task Neural Models for Translating Between Styles Within and Across Languages". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1008–1021. URL: https://aclanthology.org/C18-1086.

[Ott+19] Myle Ott et al. *fairseq: A Fast, Extensible Toolkit for Sequence Modeling*. 2019. arXiv: 1904.01038 [cs.CL].

[PT16] Ellie Pavlick and Joel Tetreault. "An Empirical Analysis of Formality in Online Communication". In: *Transactions of the Association for Computational Linguistics* 4 (2016), pp. 61–74. DOI: 10.1162/tacl_a_00083. URL: https://aclanthology.org/Q16-1005.

[RT18] Sudha Rao and Joel Tetreault. *Dear Sir or Madam, May I introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer*. 2018. arXiv: 1803.06535 [cs.CL].

[Vas+17] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].