Improving Low-Resource POS Tagging with Transfer Learning: A Case in Cantonese

Stanford CS224N Custom Project

Ting Lin
Department of Computer Science
Stanford University
linting@stanford.edu

Manh Dao
Department of Computer Science
Stanford University
manh08@stanford.edu

Trevor Carrell
Department of Computer Science
Stanford University
carrtre@stanford.edu

Abstract

Despite having over 80 million native speakers, Cantonese is a low-resource language in NLP due to being primarily a spoken form. To address problems of data scarcity, we proposed a transfer learning approach to leverage transformer-based models pretrained on Standard Chinese for Cantonese POS-tagging. We additionally compare the finetuning performance of pretrained monolingual models and multilingual models to investigate the importance of diverse language information for cross-lingual transfer. Overall, we were able to achieve a 3% increase in accuracy through finetuning.

1 Key Information to include

- Mentor: Nelson Liu
- Team Contributions: Ting implemented baselines, wrote the alignment function and worked on writeup. Trevor implemented finetuning with monolingual & multilingual models. Manh implemented finetuning with Cantonese tokenization and worked on writeup.

2 Introduction

As a classic NLP task, POS tagging is important for diverse downstream applications including dependency parsing (Nguyen and Verspoor, 2019), machine translation (Wei et al., 2024), and information extraction (Benton et al., 2021). In the past years, large language models based on the transformer architecture have achieved remarkable success on a variety of language tasks, setting high benchmarks in POS tagging for languages such as English, French and Mandarin Chinese. However, issues of data scarcity have excluded many low-resource languages from this progress, including Cantonese (Xiang et al., 2022).

To address the challenges facing Cantonese POS tagging, we draw on a rich literature of transfer learning leveraging cross-lingual similarities (Wang et al., 2019)(de Vries et al., 2022)(Mollanorozy et al., 2023). In this project, we finetuned transformer-based models pretrained on other languages, primarily Standard Chinese, with a small annotated Cantonese dataset, and evaluated the models on POS tagging accuracy tasks. We found that finetuning transformers pretrained solely on Traditional Chinese data produced better results than multilingual transformers. In addition, we investigated issues of tokenization and ran experiments using a tokenizer adapted for Cantonese.

3 Related Work

3.1 Cantonese POS Tagging

Lee et al. (2022) introduced Pycantonese, the first open-source software package for Cantonese NLP, which includes functionality for word-segmentation and POS tagging. They additionally provide access to HKCanCor, a small corpus of Cantonese conversational data that is word-segmented and annotated for part-of-speech. For word-segmentation, they use a simple longest string matching algorithm, trained on HKCanCor and rime-cantonese data . For POS-tagging, they use an averaged perceptron model, using features based on word bigrams and trigrams around a target word. These methods are motivated by a lack of large-scale, legally available Cantonese language data to support training neural models.

Huang et al. (2022) iterated upon PyCantonese by annotating a Cantonese thesaurus file from Hong Kong Cantonese Dictionary with part-of-speech tags. They established a dictionary of approximately size 6,000 for words or phrases with unique POS tags to deal with slang, allegorical speech, "wise-cracks" and common turns-of-phrases. In their schema, during POS tagging, the model first searches for the word or phrase within the POS dictionary; if not found, it is then processed by PyCantonese's pos tagging module. However, the annotated dictionary was not made open-source.

3.2 Transfer Learning for POS Tagging

Wang et al. (2019) used transfer learning from English to develop effective POS taggers and dependency parsers for Singlish, an English-based creole language. They borrow two approaches previously used for cross-annotation (Chen et al., 2016), a neural-stacking model and a neural multi-task learning model, and were able to improve state-of-the-art Singlish POS tagging accuracy. Although the paper made limited methodological innovations, and the base English model architectures used in the paper are no longer state-of-the-art (SOTA), the paper validates the use of transfer learning for syntactic knowledge. Cantonese is a low-resource language that shares many base features with a high-resource language (Mandarin) while deviating from it in both syntax and lexicon, which parallels the relationship of Singlish to English.

de Vries et al. (2022) investigated zero-shot cross-lingual transfer for POS tagging using large multilingual pre-trained models, evaluating on over 65 different source languages and 105 target languages. They found that pre-training of both source and target language, as well as matching language families, writing systems, word order systems, and lexical-phonetic distance significantly impact cross-lingual performance.

Mollanorozy et al. (2023) expands on findings that show cross-lingual transfer efficacy is strongly dependent on the similarities between the low-resource test language and the training language(s) (de Vries et al., 2022), choosing to investigate the case of Persian. They found low-resourced candidates for cross-lingual transfer from Persian using World Atlas of Language Structures, which was validated by test results. Additionally, they found that the monolingual Persian pretrained model was worse than a multilingual (XLM-RoBERTa-base) model for transfer learning, validating the importance of other languages' existence in the pre-training of the model.

4 Approach

4.1 Finetuning

Our central approach is to finetune pretrained transformers using a small Cantonese corpus for Cantonese POS Tagging. Based on their typological and genetic relationship, we posit that Standard Chinese is an ideal candidate for cross-lingual transfer to Cantonese, and this approach leverages the syntactic and lexical similarity between Standard Chinese and Cantonese while encouraging the learning of Cantonese-specific features. Motivated by Mollanorozy et al. (2023), we also want to investigate whether finetuning a multilingual model would yield better results than a monolingual Standard Chinese model.

Models For our monolingual model, we will be employing the CKIP Transformers for traditional Chinese (developed by the CKIP: Chinese Knowledge and Information Processing

group). We will specifically be using `ckiplab/bert-base-chinese-pos`, which is trained for POS-tagging respectively on the ASBC (Academia Sinica Balanced Corpus of Modern Chinese) `http://asbc.iis.sinica.edu.tw` dataset.

For our multilingual model, we will be using `bert-base-multilingual-based` Devlin

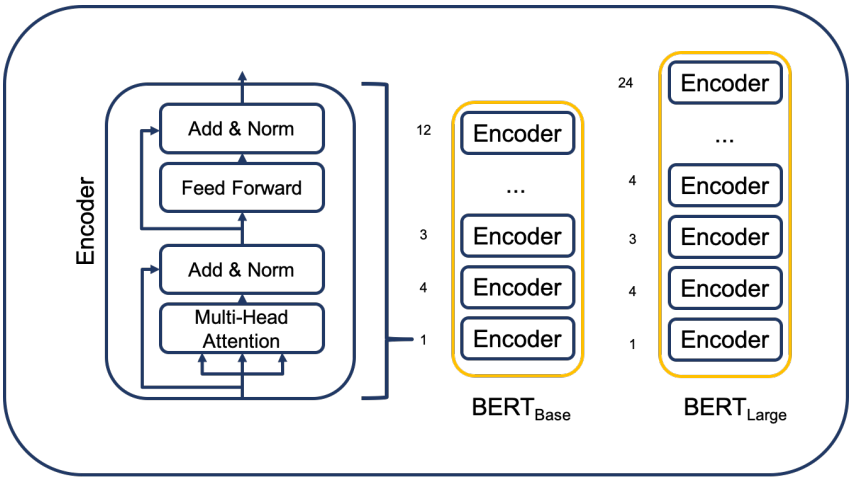


Figure 1: BERT Architecture. Both our models are BERT-base models, which include 12 encoder layers and 768 hidden units.

et al. (2019), which is the multilingual version of BERT pre-trained the top 104 languages with the largest Wikipedia using a masked language modeling (MLM) objective. We chose this model over the XLM-RoBERTa-base, the pretrained multilingual model used in de Vries et al. (2022) and Mollanorozy et al. (2023), because we want to directly compare the monolingual and multilingual models with the same architecture. The BERT-multilingual model is pretrained on both simplified Chinese and traditional Chinese.

Data The finetuning dataset we are using is the HK Cantonese Corpus (`https://github.com/fcbond/hkcancor`)(Luke and Wong, 2015). This annotated corpus consists of 230,000 Chinese words retrieved from recorded transcribed conversations between March 1997 and August 1998. We are accessing the complete corpus using PyCantonese, a Python library for Cantonese NLP (Lee et al., 2022). We will be using PyCantonese to convert the POS annotations from the CHAT format to the Universal POS scheme, allowing us to match the annotation format to the test set.

4.2 Baselines

We have three available baselines, one of which was published in a prior paper and two of which we **implemented independently** as a reproduction of results reported in a Medium article (Chulishifan, 2023). For the two reproductions, we found similar results to those previously reported.

XLM-RoBERTa base (zh) de Vries et al. (2022) finetuned XLM-RoBERTa base on the UD Chinese dataset and reported results on UD Cantonese. The UD Chinese-HK (zh-hk) dataset forms a parallel treebank to the UD Cantonese-HK (yue-hk) dataset; as such, there is some reason to expect inflated test accuracy due to the amount of lexical overlap.

12	爺爺，我做晒功課喇！	12	爺爺，我做好功課了！

Figure 1: Parallel data from UD Cantonese and UD Chinese-HK treebanks

PyCantonese PyCantonese provides a word segmentation and pos-tagging model trained on HK Cantonese Corpus (HKCanCor). We evaluate PyCantonese on the UD Cantonese-HK dataset by

tokenizing then obtaining POS labels for each sentence. We then aligned the predicted labels with the gold labels by taking the predicted label of the first subtoken character as the label for each multicharacter token in the gold set, which is the alignment method used by de Vries et al. (2022). We evaluate the aligned predictions with `poseval`, a HuggingFace evaluation library.

CKIP Bert Base We accessed `ckiplab/bert-base-chinese-pos` through HuggingFace and evaluated the model on the UD Cantonese-HK dataset using a HuggingFace token-classification pipeline. We aligned the predicted labels with the gold labels in a similar manner as previously described, and evaluate using `poseval`.

Table 1: Baseline Metrics

Model	Metrics			
	Accuracy	F1	Precision	Recall
XLM-RoBERTa base (zh)	**0.806**[1]	N/A	N/A	N/A
PyCantonese	0.736	0.725	0.729	0.735
CKIP BERT-base Chinese	0.769	**0.767**	**0.798**	**0.769**

5 Experiments

5.1 Data

Our evaluation dataset is the Universal Dependencies (UD) Cantonese HK treebank (`https://universaldependencies.org/treebanks/yue_hk/index.html`), which consists of film subtitles and of legislative proceedings of Hong Kong. It is a small dataset containing 1004 sentences and 13918 tokens, and manually annotated for UPOS, Features and Relations under the UD native scheme. Our task is to predict UPOS tags per token for each sentence in the UD Cantonese-HK treebank. All baselines were also evaluated on this dataset.

5.2 Evaluation method

Similar to our baselines, our evaluations will be made using the `poseval` library from HuggingFace, which includes accuracy and both macro-averaged and weighted f1-score, accuracy, precision, and recall for POS-tagging. We will report per-token accuracy and weighted f1-score, accuracy, precision.

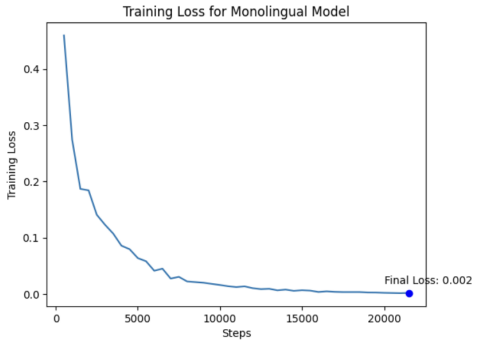
5.3 Experimental details

- **Model Configuration:** Because our pretrained monolingual model is trained to output CKIP tags instead of UD-style tags, we replace the final classifier layer of the transformer with one with dimensions corresponding to the number of UPOS tags and randomly initialized it. We also establish mappings between numerical labels and POS tags to be able to interpret model outputs and used the imported BERT-Chinese tokenizer as our tokenizer.
- **Training:** We split the HKCanCor corpus 80/20 into a training set and a dev set. We trained the model with a learning rate of 5e-5 and AdamW optimization for 20 epochs on a Google Colab GPU A100 instance, which took around 75 minutes. All training code was written using HuggingFace.

5.4 Results

For our monolingual model, we observed a 3% increase in accuracy post-finetuning compared to the pretrained model (CKIP BERT base), and significant improvement over PyCantonese, which was trained from scratch on the HkCanCor dataset. In addition, our accuracy is only 0.7% lower than that of the XLM-RoBERTa-zh base model, which was finetuned on a parallel treebank with significant overlap with the test set.

For our multilingual model, we observed a lower accuracy compared to the monolingual model. This result, which is opposite of what was found in Mollanorozy et al. (2023), could be because



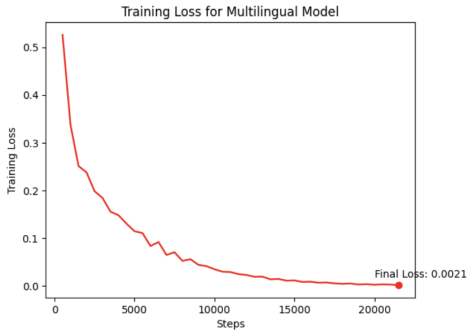


Table 2: Baseline Metrics + Our Model Results

Model	Metrics			
	Accuracy	F1	Precision	Recall
*XLM-RoBERTa base (zh)	**0.806**[2]	N/A	N/A	N/A
PyCantonese	0.736	0.725	0.729	0.735
CKIP BERT-base Chinese	0.769	0.767	0.798	0.769
Finetuned CKIP BERT-base Chinese	0.799	**0.795**	**0.799**	**0.789**
Finetuned BERT-base Multilingual	0.773	0.774	0.773	0.765

Mollanorozy et al. (2023) evaluated zero-shot cross-lingual transfer, while we are finetuning the model on the target language. It may be that access to other languages in pretraining is important for zero-shot transfer, but not for cross-lingual transfer with finetuning. It could also be because the quantity difference in pretraining data of the closest orthographic language (Traditional Chinese) is larger between our models than it is in the Persian models used by Mollanorozy et al. (2023).

6 Analysis

In our analysis, we focus on the model which produced better results, which is the monolingual model.

6.1 Model Errors

When examining the F1 score for each individual UD POS tag, we find that the PUNCT, PRON and INTJ had the highest F1 scores, and DET, SCONJ, and CCONJ had the lowest F1 scores. Strikingly, the F1 score for both DET and SCONJ was 0.0, showing that our model was completely unable to handle these two part-of-speech categories.

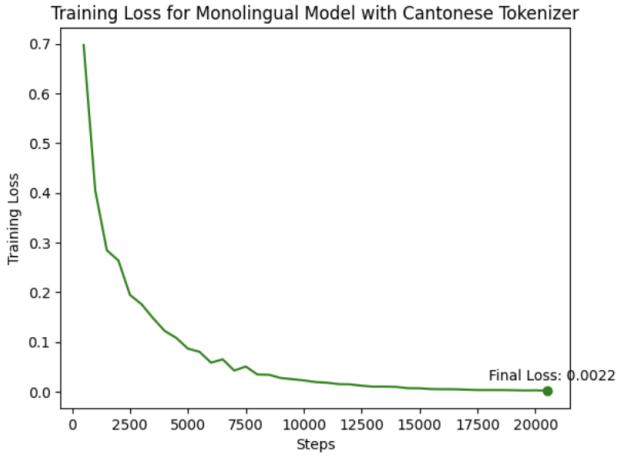
This could be because both DET and SCONJ are very infrequent POS categories. Within our dataset, DET tokens only make up 2% of the total tokens, and SCONJ tokens only make up 1%. However, this does not fully explain the failure to handle these two particular tags, since although CCONJ also only make 1% of all tokens, it has a F1 score of 0.393.

The answer could lie in the differential treatment of DET tokens and SCONJ tokens in the finetuning and testing dataset. For example, “如果” (if) is treated in the finetuning dataset as a CCONJ token, while it is the most common SCONJ token in the testing dataset. “去” (go) is the third most common SCONJ token in the testing dataset, but is overwhelmingly annotated in the finetuning dataset as a VERB.

It could also be that many DET and SCONJ tokens are "confusable" even within the testing dataset; for example, the three most common DET tokens in UD-Cantonese are also frequently annotated as other categories: “呢”(this)(PART 328, DET 75, VERB 3, NOUN 1), “依個”(DET 25, PRON 9)(that one), “呢個”(this one)(PRON 20, DET 16, PART 2).

6.2 Tokenization

We find that one of the primary problems with the model is that all of output text is tokenized into single-character subtokens, even in situations where two or more characters should be treated as a unit. As an example, “呢的” is tokenized into “呢”(this) and “的”(plural particle). This is likely because the tokenizer used is `google-bert/bert-base-chinese`, which is ill-suited for Cantonese tokenization. For one, its vocab is in simplified Chinese, whereas Cantonese is usually written using traditional Chinese. Second, it lacks characters that are unique to Cantonese. These characters are treated as out-of-vocab items and mapped to the [’UNK’] token. Third, there are some characters that are shared between Chinese and Cantonese, but have different meanings - for example, “晒” means "to expose to the sun" in Chinese, but means "finished" in Cantonese. Although we remedy some of these problems during evaluation by aligning the predicted tokens to the gold tokens, the discrepency between the underlying linguistically meaningful units in our Cantonese data and the Chinese tokenizer is likely a barrier towards achieving better POS tagging accuracy.



We considered two methods to address this issue. First, we considered training a Cantonese tokenizer from scratch using our finetuning corpus data. However, due to the small corpus size, this will likely perform weakly compared to the pretrained tokenizer. This is amplified by the fact that there is some ambiguity in Chinese segmentation and tokenization due to its orthography (literature has shown that word intuition agreement is around 90% amongst native Chinese speakers Wang et al. (2017)), and the tokenization in different annotated datasets are not in strict agreement. Therefore, a tokenizer trained on less data would be less robust and suited to handle these discrepencies.

Secondly, we considered adding all out-of-vocab tokens from our finetuning dataset to the pretrained tokenizer, and initializing random embeddings for the newly added tokens. Luckily, we were able to find an existing implementation of a Cantonese tokenizer that achieved that objective (`https://github.com/ayaka14732/bert-tokenizer-cantonese`). `BERT-tokenizer-cantonese` was created by 1) converting the tokens of the original tokeniser from Simplified Chinese to Traditional Chinese, while keeping the corresponding embeddings fixed; 2) adding 150 Cantonese-specific characters to the tokenizer vocabulary (Ayaka14732, 2024). We conducted an additional monolingual experiment with `ckiplab/bert-base-chinese-pos` as our model and `BERT-tokenizer-cantonese` as the tokenizer, and trained with the same model configurations, hyperparameters, GPU instance and number of epochs.

Somewhat surprisingly, we found that this did not yield any improvements. It could be the case that the 150 Cantonese-specific characters were not the biggest issue for tokenization, or that simply adding them to the vocab was insufficient, as it still did not encourage the recognition of Cantonese multi-character words. It could also be an artifact of our training parameters, as the model with Cantonese tokenization yielded a slightly higher final loss. While we were not able to investigate further for this project, this should be examined in future work.

Table 3: Comparison between Finetuned Models with and without Cantonese Tokenizer

Model	Metrics			
	Accuracy	F1	Precision	Recall
Finetuned CKIP BERT-base Chinese	**0.799**	**0.795**	**0.799**	**0.789**
Canto-tokenized Finetuned CKIP BERT-base Chinese	0.752	0.743	0.746	0.752

7 Conclusion

In this project, we finetuned both monolingual standard Traditional Chinese transformers and multilingual transformers using Cantonese data. We were able to observe an improvement of 3% in testing accuracy over the baseline for our best model, which is close to state-of-the-art (SOTA) results currently achieved on the UD-Cantonese dataset [we also note again that the current SOTA model was finetuned on a parallel treebank]. In our experiments, finetuning the monolingual model yielded better results than the multilingual model, highlighting the importance of quantity of orthographically-similar pretraining data for cross-lingual transfer. We additionally discussed and investigated issues of tokenization, although we did not find improvements when using a tokenizer adapted for Cantonese data.

There are several limitations to this project. First, we did not perform hyperparameter tuning, and opted for the default hyperparameters as defined by the HuggingFace training arguments class. This may have limited our model performance. Second, our finetuning and testing dataset are quite different in domain, and they were natively annotated in different POS schemes, which had to be mapped to consistent UD style tags. While being able to achieve good cross-domain performance would be a sign of robustness, this evaluation scheme also limits our ability to directly assess within-domain model performance. The necessity to map POS tags from different schemes to a unified format introduces potential errors and inconsistencies, which were partially analyzed in section 6.1.

Future work should consider 1) further examining tokenization issues, including pretraining a tokenizer from scratch on a sufficiently large corpus, 2) further pretraining a monolingual Traditional Chinese model with some multilingual data, to see if access to additional languages improves cross-lingual transfer performance once quantity of Traditional Chinese pretraining data is controlled for. If more monolingual Traditional Chinese models with different architectures are released, similar experiments should be conducted to see if the SOTA performance can be improved.

References

Ayaka14732. 2024. bert-tokenizer-cantonese. `https://github.com/ayaka14732/bert-tokenizer-cantonese`. 2024/03/25.

Adrian Benton, Hanyang Li, and Igor Malioutov. 2021. Cross-register projection for headline part of speech tagging. *CoRR*, abs/2109.07483.

Hongshen Chen, Yue Zhang, and Qun Liu. 2016. Neural network for heterogeneous annotations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 731–741.

Chulishifan. 2023. Evaluating cantonese performance in nlp systems. *Medium*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Chunxiao Huang, Chunyu Li, Shaowen Yao, Ye Ding, Mingsuo Bao, and Kun She. 2022. A hybrid scheme for parsing cantonese text based on pycantonese plus and pyltp. In *2022 European Conference on Natural Language Processing and Information Retrieval (ECNLPIR)*, pages 47–51.

Jackson L. Lee, Litong Chen, Charles Lam, Chaak Ming Lau, and Tsz-Him Tsui. 2022. Pycantonese: Cantonese linguistics and nlp in python. In *Proceedings of The 13th Language Resources and Evaluation Conference*. European Language Resources Association.

Kang-Kwong Luke and May LY Wong. 2015. The hong kong cantonese corpus: design and uses. *Journal of Chinese Linguistics*, pages 309–330.

Sepideh Mollanorozy, Marc Tanti, and Malvina Nissim. 2023. Cross-lingual transfer learning with Persian. In *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 89–95, Dubrovnik, Croatia. Association for Computational Linguistics.

Dat Quoc Nguyen and Karin Verspoor. 2019. From pos tagging to dependency parsing for biomedical event extraction. *BMC bioinformatics*, 20:1–13.

Wietse de Vries, Martijn Wieling, and Malvina Nissim. 2022. Make the best of cross-lingual transfer: Evidence from pos tagging with over 100 languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7676–7685.

Hongmin Wang, Jie Yang, and Yue Zhang. 2019. From genesis to creole language: Transfer learning for singlish universal dependencies parsing and pos tagging. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(1).

Shichang Wang, Chu-Ren Huang, Yao Yao, and Angel Chan. 2017. Word intuition agreement among chinese speakers: a mechanical turk-based study. *Lingua Sinica*, 3:1–18.

Chengwei Wei, Runqi Pang, and C-C Jay Kuo. 2024. Gwpt: A green word-embedding-based pos tagger. *arXiv preprint arXiv:2401.07475*.

Rong Xiang, Hanzhuo Tan, Jing Li, Mingyu Wan, and Kam-Fai Wong. 2022. When cantonese nlp meets pre-training: Progress and challenges. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: Tutorial Abstracts*, pages 16–21.