Low Resource Similar Language Neural Machine Translation for Tamil-Telugu

Vandan Mujadia and Dipti Misra Sharma
Machine Translation - Natural Language Processing Lab
Language Technologies Research Centre
Kohli Center on Intelligent Systems
International Institute of Information Technology - Hyderabad
vandan.mu@research.iiit.ac.in, dipti@iiit.ac.in

Abstract

This paper describes the participation of team oneNLP (LTRC, IIIT-Hyderabad) for the WMT 2021 task, similar language translation[1]. We experimented with transformer based Neural Machine Translation and explored the use of language similarity for Tamil-Telugu and Telugu-Tamil. We incorporated use of different subword configurations, script conversion and single model training for both directions as exploratory experiments.

1 Introduction

Machine Translation (MT) is a field of Natural Language Processing which aims to translate a text from one natural language to another. The meaning of the source text must be fully preserved in the resulting translated text in the target language. Recent years have seen significant quality advancements in machine translation with the advent of Neural Machine Translation. For the translation task, different types of machine translation systems have been developed and they are mainly categorized into Rule based Machine Translation (RBMT)(Forcada et al., 2011), Statistical Machine Translation (SMT) (Koehn, 2009) and Neural Machine Translation (NMT) (Bahdanau et al., 2014).

Neural machine translation (NMT) shows high quality in terms of output fluency and translation quality, when large amounts of parallel data are available (Barrault et al., 2020). Unfortunately, for most language pairs, parallel data is either scare or non-existent. To overcome this, unsupervised MT (UMT) (Artetxe et al., 2020) focuses on utilising monolingual data to generate synthetic parallel training data. Other techniques like back-translation(Sennrich et al., 2015),(Hoang et al., 2018), (Feldman and Coto-Solano, 2020) or denoising(Kim et al., 2019) also rely on parallel corpora of other language pairs and/or large quantities of monolingual data.

This paper describes our experiments for very low resourced similar language translation. For our work, we focused only on Tamil-Telugu language pair (both directions) and participated in a constrained setting.

We experimented only with Transformer (Vaswani et al., 2017) based Neural Machine Translation throughout. Along with it, to tackle high agglutination of both languages, we explored the morph (Virpioja et al., 2013) induced sub-word segmentation with byte pair encoding (BPE)(Sennrich et al., 2016).

Similar to Multilingual Neural Machine Translation (MNMT), we explored the use of a tag trick, where a token like "< 2xx >" (xx is language code) is prefixed to each source sentence to indicate the desired target language(Dabre et al., 2020). Here, we trained a single model for both directions (Tamil-Telugu and Telugu-Tamil) on given parallel data and monolingual data under MNMT setting.

The sections of the paper are organised as following: Section 2 describes Data, Section 3 and 4 describe pre-processing and Training Configuration and in Section 5 we talk about results and we conclude in section 6.

2 Data

We utilised provided parallel corpora for Tamil<->Telugu MT task. Apart form parallel corpus, we randomly selected 0.1M monolingual corpora from IndicCorp monolingual corpus[2] for Tamil and Telugu. Table-1 describes the training and development data (parallel and monolingual) used in all our experiments under constrained setting.

[1] https://www.statmt.org/wmt21/similar.html

[2] https://indicnlp.ai4bharat.org/corpora/

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 288–291
November 10–11, 2021.

Data	Sents	Token	Type
Train			
Tamil (Parallel)	40,147	0.68M	74K
Telugu (Parallel)	40,147	0.72M	90K
Development			
Tamil (Parallel)	1261	29K	9K
Telugu (Parallel)	1261	30K	10K
Tamil (Mono)	0.1M	-	-
Telugu (Mono)	0.1M	-	-

Table 1: Tamil-Telugu WMT2021 Training data

3 Data Pre-Processing

To tokenize and clean both Tamil and Telugu corpora (train, test, valid and monolingual), we used IndicNLP Tool[3] with in-house tokenizer as a first step. Following subsections explain other pre-processing steps of experiments.

3.1 Morph + BPE Segmentation

Based on token/type ratio, both Tamil and Telugu are morphologically rich languages from Table-1. Translating from (and to) morphologically-rich agglutinative language is more difficult due to their complex morphology and large vocabulary. We address this issue with morphology and BPE(Sennrich et al., 2016) based segmentation method as prescribed in (Mujadia and Sharma, 2020). We utilized unsupervised Morfessor (Virpioja et al., 2013) by training it on monolingual data of Tamil and Telugu. We then applied this trained Morfessor model on our corpora (train, test, development) to get meaningful stem, morpheme, suffix segmented sub-tokens for each word in a sentence. Subsequently, we applied the subword algorithm on top of the morph segmentation and used the derived sequence in training.

3.2 Training as Multilingual Neural Machine Translation (MNMT)

As an exploratory experiment, we configure a similar low resource machine translation problem as a multilingual machine translation problem. For both translation directions (Tamil-Telugu and Telugu-Tamil) we trained a single model to take advantage of language similarity among these languages. First, we converted both languages into Roman script using litcm[4]. Second, we prefixed “<2TE>” for Tamil to Telugu and “<2TA>” for Telugu to Tamil to the respective source sentences. Apart from this, we also utilised monolingual data as a monolingual translation. For this we prefixed “<2TE>” for Telugu to Telugu and “<2TA>” for Tamil to Tamil translation.

4 Training Configuration

Throughout all experiments, we used Transformer sequence to sequence architecture with the following configuration.

- Morph + BPE based subword segmentation, Embedding size : 512 Transformer for encoder and decoder, rnn_size 512, heads 4 encoder - decoder layers : 2, label smoothing : 1.0, dropout : 0.30, Optimizer : Adam, Beam size : 4 (train) and 10 (test), training steps : 20K

For these experiments, we used shared vocab across trainings. We used Opennmt-py (Klein et al., 2020) toolkit with above configuration for our experiments.

Using the above described pre-processing and configuration, we performed experiments on word level, BPE level and morph + BPE level for input and output. The results are discussed in following Result section.

5 Result

Feature	**BPE**	**Dev**
Script Conversion (ta to te)	-	0.57
Word	-	5.12
BPE	20K	6.07
Morph + BPE	20K	6.25
Morph + BPE (MNMT)	20K	6.65

Table 2: BLEU scores for Tamil-Telugu on Development set. BPE stands for byte pair encoding (subword), Morph for Morphological segment and MNMT for Multilingual Neural Machine Translation based method as discussed in Section-3.2

Table-2 and Table-3 show performance of our systems with different configurations in terms of BLEU score (Papineni et al., 2002) for Tamil-Telugu and Telugu-Tamil respectively on the development data. To get trivial, non-translation baseline, we used aksharamukha[5] script conversion

[3]http://anoopkunchukuttan.github.io/indic nlp library/

[4]https://github.com/irshadbhat/litcm

[5]https://aksharamukha.appspot.com/converter

Feature	BPE	Dev
Script Conversion (te to ta)	-	0.41
Word	-	5.72
BPE	20K	6.37
Morph + BPE	20K	6.45
Morph + BPE (MNMT)	20K	6.76

Table 3: BLEU scores for Telugu-Tamil on Development set. BPE stands for byte pair encoding (subword), Morph for Morphological segment and MNMT for Multilingual Neural Machine Translation based method as discussed in Section-3.2

tool to convert script from Tamil-Telugu (both direction). We achieved highest 6.65 and 6.76 development and 3.67 and 5.03 test BLEU scores for Tamil-Telugu and Telugu-Tamil systems respectively (all are of MNMT based systems).

Table-2 and Table-3 show that non-translation baselines are also low in terms of BLEU scores which indicates that the task much harder even though languages are similar. The results show that for low resource settings, transformer network based MT models can be improved with morph based segmentation along with byte pair encoding for morph rich languages. Also, forming it as a Multilingual machine translation problem, along with monolingual data, it improves the quality of MT models. This may be due to language similarity and use of monolingual data, as it is helping models to do better generalization by learning better source language encoding and target language fluency.

6 Conclusion

From our experiments, we conclude that linguistic feature such as morph based segmentation with subword segments along with MNMT is a promising approach for similar language translation.

References

Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre. 2020. A call for more rigor in unsupervised cross-lingual learning. *arXiv preprint arXiv:2004.14958*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Yvette Graham, Paco Guzman, et al. 2020. Proceedings of the fifth conference on machine translation. In *Proceedings of the Fifth Conference on Machine Translation*.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Computing Surveys (CSUR)*, 53(5):1–38.

Isaac Feldman and Rolando Coto-Solano. 2020. Neural machine translation models with back-translation for the extremely low-resource indigenous language bribri. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3965–3976.

Mikel L Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine translation*, 25(2):127–144.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24.

Yunsu Kim, Jiahui Geng, and Hermann Ney. 2019. Improving unsupervised word-by-word translation with language model and denoising autoencoder. *arXiv preprint arXiv:1901.01590*.

Guillaume Klein, François Hernandez, Vincent Nguyen, and Jean Senellart. 2020. The opennmt neural machine translation toolkit: 2020 edition. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (AMTA 2020)*, pages 102–109.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Vandan Mujadia and Dipti Misra Sharma. 2020. Nmt based similar language translation for hindi-marathi. In *Proceedings of the Fifth Conference on Machine Translation*, pages 414–417.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, et al. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline.