# A3-108 Machine Translation System for Similar Language Translation Shared Task 2021

**Saumitra Yadav, Manish Shrivastava**
Machine Translation - Natural Language Processing Lab
Language Technologies Research Centre
Kohli Center on Intelligent Systems
International Institute of Information Technology - Hyderabad
saumitra.yadav@research.iiit.ac.in
m.shrivastava@iiit.ac.in

## Abstract

In this paper, we describe our submissions for the Similar Language Translation Shared Task 2021. We built 3 systems in each direction for the Tamil $\Longleftrightarrow$ Telugu language pair. This paper outlines experiments with various tokenization schemes to train statistical models. We also report the configuration of the submitted systems and results produced by them.

## 1 Introduction

Machine translation is a process of translating text from a source to a target language. There are multiple ways of building such a system - Rule-based, Data-driven, Hybrid etc. In this shared task, we use data-driven method to create machine translation system for Tamil $\Longleftrightarrow$ Telugu. Due to low-resource setting of this language pair in the shared task, we use Statistical Machine translation method (Koehn et al., 2003),(Koehn and Knowles, 2017) to build systems.

Tamil Telugu language pair comes under the bracket of similar languages. Similar languages show similarity in their lexical and syntactical properties (Kunchukuttan et al., 2014a). This may be due to them being in close proximity of each other for long time. This can also be due to common ancestry. In the current digital context, translation between similar languages is of importance. But there can be scarcity of good quality parallel text. In the current shared task, we have a language pair which is morphologically rich and with $\simeq$39K parallel sentences. So, following Kunchukuttan and Bhattacharyya (2017) and Kunchukuttan et al. (2014b) we use sentencepiece[1] (Kudo and Richardson, 2018) and morfessor[2] (Virpioja et al., 2013) to segment tokens in the dataset into subwords. And due to the size of parallel text ($\simeq$39K parallel text) coming under purview of low resource, we make use of Moses[3](Koehn et al., 2007) to create statistical machine translation models(Koehn and Knowles, 2017).

For this shared task we developed 3 translation systems (1 Primary and 2 Contrastive) in each direction Tamil $\Longleftrightarrow$ Telugu. For each output we post-processed and detokenized translation output depending on the tokenization scheme for target language. To choose a primary and 2 contrastive systems, we compared BLEU (Papineni et al., 2002) scores on output of development dataset for each system using sacrebleu[4] (Post, 2018). The Following sections give more details about the systems developed.

## 2 SMT systems using different schemes

We used various tokenization schemes to build translation systems. Evaluated these systems on the development dataset. After post-processing, detokenizing and scoring each translation output, we submit output systems as primary and contrastive submissions accordingly.

### 2.1 Data and preprocessing

We used parallel data provided by the organizers to train all the models. IndicNLP[5] (Kunchukuttan, 2020) was used to normalize and tokenize datasets. 2 Subword models were trained on tokenized text for each language. Sentencepiece(Kudo and Richardson, 2018) was used to prepare a subword tokenizer model with vocabulary size set to 32000 and character coverage set to 0.9995. Another alternative tokenization model was trained on morfessor(Virpioja et al., 2013). To create 3 systems for each translation direction, we used the

[1] https://github.com/google/sentencepiece
[2] https://github.com/aalto-speech/morfessor
[3] https://github.com/moses-smt/mosesdecoder
[4] https://github.com/mjpost/sacrebleu
[5] https://github.com/anoopkunchukuttan/indic_nlp_library



*Proceedings of the Sixth Conference on Machine Translation (WMT)*, pages 304–306
November 10–11, 2021. 

| Dataset with tokenization | Tamil | | | Telugu | | | Total number of Lines |
|---|---|---|---|---|---|---|---|
| | Total Token Count | Total Unique Token | Avg Token Per line | Total Token Count | Total Unique Token | Avg Token Per line | |
| Train.basicTok | 691433 | 74341 | 17.22 | 725365 | 72949 | 18.06 | 39836 |
| Dev.basicTok | 30017 | 9683 | 23.80 | 30359 | 9467 | 24.07 | 1261 |
| Train.spm | 770632 | 31674 | 19.63 | 956023 | 31782 | 24.35 | 39246 |
| Dev.spm | 36672 | 8647 | 29.08 | 41779 | 9112 | 33.13 | 1261 |
| Train.morf | 956485 | 13956 | 24.47 | 947463 | 17823 | 24.24 | 39081 |
| Dev.morf | 45279 | 5496 | 35.90 | 43602 | 6380 | 34.57 | 1261 |

Table 1: Statistics of Tamil and Telugu datasets

following tokenization schemes,

- basicTok: bitext is tokenized with IndicNLP.
- morf: each training file in the parallel text is tokenized into subwords with the respective morfessor model.
- spm: each training file in the parallel text is tokenized into subwords with the respective sentencepiece model

Table 1 shows the statistics of the Tamil and Telugu dataset for each tokenization scheme after using `clean-corpus-n.perl` script with 1,70 as min,max line length for training text. No additional monolingual dataset was used in building any of the models.

### 2.2 MT Systems

We build a trigram language model with kneser ney smoothing for each language in each tokenization scheme using KenLM (Heafield, 2011). And used Moses (Koehn et al., 2007) to train an SMT system. MERT (Och, 2003) is used for tuning the trained model on development datasets. The performance of all systems, for each language direction on respective tokenized development datasets, is given in Table 2. For this shared task, we submit 3 systems (1 PRIMARY and 2 CONTRASTIVE) for each language direction for evaluation. Depending on scores on development dataset, systems build were submitted as,

| | Tamil ->Telugu | Telugu ->Tamil |
|---|---|---|
| basicTok | 7.7 | 9.9 |
| spm | 5.2 | 9.0 |
| morf | 7.7 | 9.8 |

Table 2: BLEU score on development dataset for each system

- For Telugu to Tamil,
  - A3-108_TE_TA_PRIMARY.txt: basicTok Telugu -> basicTok Tamil system - trained using SMT model - tokenized using indic nlp library.
  - A3-108_TE_TA_CONTRASTIVE1.txt: morf Telugu -> morf Tamil system - trained using SMT model - tokenized using morfessor into subwords for training
  - A3-108_TE_TA_CONTRASTIVE2.txt: spm Telugu -> spm Tamil system - trained using SMT model - tokenized using sentencepiece into subwords for training
- For Tamil to Telugu,
  - A3-108_TA_TE_PRIMARY.txt: morf Tamil -> morf Telugu system - trained using SMT model - tokenized using morfessor into subwords for training
  - A3-108_TA_TE_CONTRASTIVE1.txt: basicTok Tamil -> basicTok Telugu system - trained using SMT model - tokenized using indic nlp library.
  - A3-108_TA_TE_CONTRASTIVE2.txt: spm Tamil -> spm Telugu system - trained using SMT model - tokenized using sentencepiece into subwords for training

### 2.3 Results

This subsection compares the results of our systems, which we received from organizers, in terms of BLEU scores. Table 3 shows the BLEU scores for Telugu to Tamil systems. In comparison with other systems, all of our system outputs score highest. We were hoping that, in test cases, models using subwords for training and translating would prove to be better than basicTok, but that was not the case. Instead models trained on basicTok fared better.

| System Type | BLEU | RIBES | TER |
|---|---|---|---|
| PRIMARY (basicTok) | 8.37 | 43.55 | 95.884 |
| CONTRASTIVE1 (morf) | 7.89 | 46.24 | 95.627 |
| CONTRASTIVE2 (spm) | 7.43 | 42.54 | 94.964 |

Table 3: Scores on test dataset for each Telugu to Tamil system

Table 4 shows the BLEU score we received for Tamil to Telugu systems. Our system outputs from

| System Type | BLEU | RIBES | TER |
|---|---|---|---|
| CONTRASTIVE1 (basicTok) | 5.54 | 40.58 | 98.082 |
| PRIMARY (morf) | 5.23 | 42.37 | 98.662 |
| CONTRASTIVE2 (spm) | 3.32 | 34.42 | - |

Table 4: Scores on test dataset for each Tamil to Telugu system

CONTRASITVE1 and PRIMARY submission are in the top 3 in comparison with other systems. Here again, we see basicTok model fared a bit better than model trained on morf segmented dataset. And sentencepiece model was $\simeq$2 BLEU points behind both the systems. These BLEU scores (CONTRASTIVE1, PRIMARY) are in the top 3. Again, we were hoping, that in test cases, models using subwords for training and translating would prove to be better. But as was case in Telugu to Tamil, here also models trained on basicTok dataset fared better, followed by models trained on morfessor segmented dataset.

## References


Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Anoop Kunchukuttan. 2020. The IndicNLP Library. `https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf`.

Anoop Kunchukuttan and Pushpak Bhattacharyya. 2017. Learning variable length units for SMT between related languages via byte pair encoding. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 14–24, Copenhagen, Denmark. Association for Computational Linguistics.

Anoop Kunchukuttan, Abhijit Mishra, Rajen Chatterjee, Ritesh Shah, and Pushpak Bhattacharyya. 2014a. Shata-anuvadak: Tackling multiway translation of Indian languages. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1781–1787, Reykjavik, Iceland. European Language Resources Association (ELRA).

Anoop Kunchukuttan, Ratish Pudupully, Rajen Chatterjee, Abhijit Mishra, and Pushpak Bhattacharyya. 2014b. The iit bombay smt system for icon 2014 tools contest. *NLP Tools Contest at ICON 2014*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

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