# The TALP-UPC System Description for WMT20 News Translation Task: Multilingual Adaptation for Low Resource MT

**Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa,**
{carlos.escolano,marta.ruiz,jose.fonollosa}@upc.edu,
TALP Research Center
Universitat Politècnica de Catalunya, Barcelona

## Abstract

In this article, we describe the TALP-UPC participation in the WMT20 news translation shared task for Tamil-English. Given the low amount of parallel training data, we resort to adapt the task to a multilingual system to benefit from the positive transfer from high resource languages. We use iterative back-translation to fine-tune the system and benefit from the monolingual data available. In order to measure the effectivity of such methods, we compare our results to a bilingual baseline system.

## 1 Introduction

Modern NMT systems such as Transformer require large amounts of training data in order to obtain good generation results. For this reason, low resource languages represent a good opportunity to explore new techniques to treat data more efficiently and benefit from available sources of data like monolingual corpora.

From the WMT20 news tasks proposed languages we are presenting our results on the English-Tamil language pair, Tamil is an official language from India, Sri Lanka, and Singapore having approximately 75 million native speakers. It belongs to the Dravidian family, originated in Asia.

Two principal reasons can make Tamil a challenging language for machine translation: script and agglutination. Tamil's script consists of 12 vowels and 18 consonants plus one special character, allowing the combination of 247 possible characters. Compared to the Latin script employed by most western languages, it is an order of magnitude higher in the number of possible characters.

Also, by agglutination, suffixes can be added to root words to form new ones. These words can lead to multiple words in the target language in the context of machine translation, which may affect attention and decoding in NMT systems.

This work discusses the system proposed for the evaluation in which we combine the use of multilingual parallel data with monolingual data to boost the performance of our proposed NMT system.

## 2 Low Resource NMT

Modern NMT systems benefit from having hundreds of thousands or even millions of parallel sentences. When working with low resource language pairs, the two main approaches are the use of monolingual corpora and multilingual NMT. While parallel data may be difficult to obtain for low resource languages, monolingual data is usually more available, as it does not require any additional labeling.

A common approach to benefit from monolingual data is back-translation (Sennrich et al., 2016a), which consists of translating a monolingual corpus to generate synthetic corpora that can be later employed to continue training. Similar techniques create a synthetic pseudo-parallel corpus through a pivot language (Casas et al., 2019) that can be then trained similarly to back-translation when data is available between the desired language pair and a pivot high resource language. More recently, iterative back-translation (Hoang et al., 2018) was proposed. This technique allows the system to generate synthetic data while updating the system, so better the new data improves as the system trains. On the other hand, several works on Multilingual NMT have shown benefits for low resource language pairs by allowing positive transfer from the high resource languages, boosting the performance of the low resource ones. Different architectures have been proposed that show this behavior, from universal models where all parameters are shared between all languages (Johnson et al., 2017), to architectures that share a common device that maps representations into a shared represen-

*Proceedings of the 5th Conference on Machine Translation (WMT)*, pages 134–138
Online, November 19–20, 2020.

tation space (Firat et al., 2016; Zhu et al., 2020), to architectures that do not share parameters (Escolano et al., 2019; Escolano et al.; Schwenk and Douze, 2017).

In the context of the WMT20 Tamil-English news shared task, as the provided parallel data is limited, we resorted to a combination of both proposed methods by incrementally train the new language pair into a Multilingual NMT system using the provided parallel data, to later fine-tune the system using iterative-back-translation with monolingual corpora.

## 3 Related Work

Previous works (Choudhary et al., 2018) have shown that Indian languages are usually a challenge for NMT systems due to their difference in terms of vocabulary and grammar compared to western languages such as English. Also, standard preprocessing methods do not always work well with them, so specific solutions are required to obtain good results.

In the context of NMT, previous systems, such as MIDAS (Choudhary et al., 2018), proved that the use of subword units leads to significant improvements in translation quality when applied to Tamil by preventing Out of Vocabulary words in at generation time.

## 4 Corpora and Data Preparation

All proposed systems in this work are constrained using exclusively data provided by the task's organization. The multilingual initial system was trained using *Europarl v8*, for all translation directions between English, French, Spanish, and German. For English-Tamil *PMIndia*, *Tanzil v1*, *The UFAL EnTam corpus*, The *NLPC UOM En-Ta corpus*, *Wikimatrix*, and *Wikitiles*. As monolingual Tamil data, we used News Crawl, while for English, we used *News-commentary*.

We processed all non-Tamil data following *Moses* (Koehn et al., 2007) scripts provided by the organization. For each one, we applied punctuation normalization, tokenization, and true-casing. Then each language is independently tokenized using BPE (Sennrich et al., 2016b) with 32 thousand operations. Table 1 the estatistics for each language. Tamil data has been tokenized at word-level using *Indic-NLP* (Kunchukuttan, 2020) and then tokenized with BPE with 16 thousand operations.

| corpus | lang | sentences | words |
|---|---|---|---|
| DE-EN | DE | 1758872 | 40265543 |
| | EN | 1758872 | 40265543 |
| DE-ES | DE | 1663458 | 37698204 |
| | ES | 1663458 | 40808518 |
| DE-FR | DE | 1681466 | 37410662 |
| | FR | 1681466 | 43056346 |
| EN-ES | EN | 1769606 | 41803882 |
| | ES | 1769606 | 43156309 |
| EN-FR | EN | 1770112 | 41211543 |
| | FR | 1770112 | 45196313 |

Table 1: Corpus statistics in number of words and sentences for the language pairs of the Multilingual initial system.

| corpus | lang | set | sentences | words |
|---|---|---|---|---|
| EN-TA | EN | train | 494310 | 7355160 |
| | | test | 1275 | 29774 |
| | TA | train | 494310 | 15163570 |
| | | test | 1275 | 66564 |
| EN | EN | train | 608912 | 14995557 |
| TA | TA | train | 504320 | 6426186 |

Table 2: Corpus statistics in number of words and sentences for the English-Tamil parallel data and English and Tamil monolingual training sets.

Table 2 show the statistics for the parallel Englist-Tamil data as well as the monolingual corpora used.

As test set, we used 1275 lines extracted from the development set provided from the organization, keeping the remaining ones as validation set.

## 5 System Description

In this section, we are going to discuss the details of the pipeline followed to create the translations systems for this submission, including the multilingual supervised pretraining and the unsupervised fine-tuning using monolingual corpora.

### 5.1 Multilingual Supervised Pretraining

**Methodology.** Following the proposed model in (Escolano et al., 2020), new languages can be added to the system without retraining the system, just using parallel data to one of the initial ones. In this work, we added Tamil using the provided parallel data to English. To train the new Tamil to English translation direction, a new Tamil encoder is added to the system with the previous English encoder frozen, to prevent the model from affecting the performance of the remaining pairs. Training with the

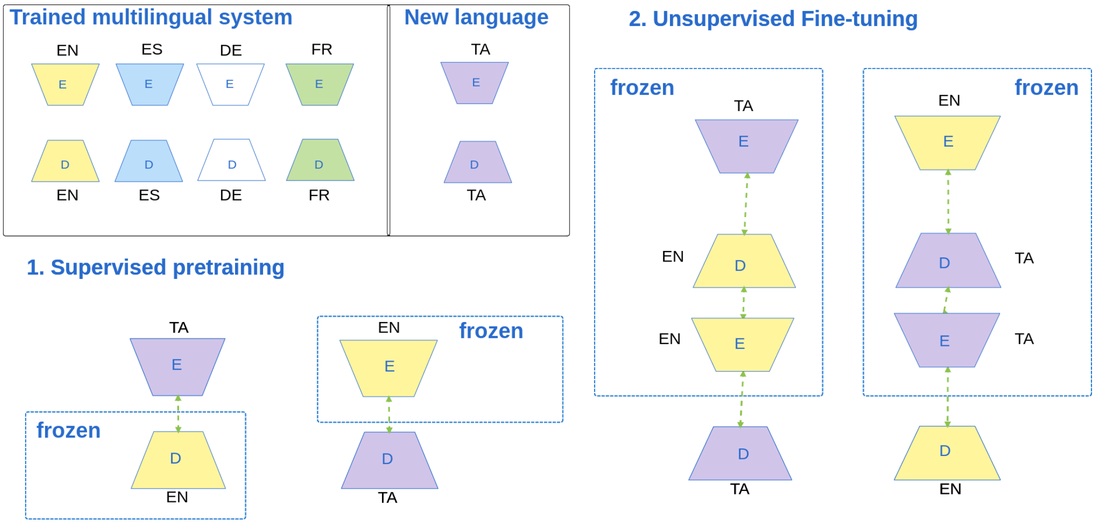

Figure 1: Training pipeline. Step 1 Supervised preptraining, Step 2 Unsupervised fine-tuning.

frozen decoder induces the new encoder to learn a similar representation to the ones already in the multilingual model. In addition, as the English decoder has been trained with more data from all the language pairs in the Multilingual NMT system, we have positive transfer from the frozen modules to the new ones, boosting the translation performance compared to the bilingual NMT baseline. Following the same principles, the English-Tamil translation direction is trained by freezing the English encoder and training the Tamil decoder to force the shared representation. In this case, we also notice the positive transfer compared to the baseline trained with just parallel data. See in Figure 1 the schema of the supervised pretraining that we have just described.

**Implementation.** For this work, all encoders and decoders were implemented using the Transformer (Vaswani et al., 2017) architecture, with 6 layers, 8 heads, 512 embedding size, and 2048 feed-forward size for each of them, and everything was implemented using *Fairseq*'s(Ott et al., 2019) 0.6 release. The multilingual NMT model was trained in a single NVIDIA TITAN XP for 50 thousand updates using adam optimizer with 0.001 as learning, 4000 warmup updates and updating every 16 batches of 2000 tokens. Adding Tamil-English and English-Tamil directions to the system took approximately 45 thousand updates using the same parameters and GPU configuration.

### 5.2 Monolingual Unsupervised Fine-tuning

**Methodology.** The previous process has benefited from the additional corpus from the Multilingual NMT system, but as stated before, monolingual data is another common source of improvement for NMT systems. In this section, we are going to discuss how we added monolingual data to the previously described model. To employ the monolingual data available in our system, we define an autoencoder using the already trained encoder and decoder modules in the given language. These modules are not trained to regenerate the input, we introduce an adaptor, between both modules, responsible for processing the representation generated and output a new one understood by the decoder. Taking advantage of the architecture, we can use one of the decoders to greedy decode the representation created and encode it back with one the encoders, to compute the reconstruction of the monolingual input. Figure 1 showcases in ”unsupervised fine-tunning” how is process is applied in our work to use both Tamil monolingual data with an English adaptor and English data with the Tamil adaptor.

In this work, both encoder and adaptor were frozen, and only the final decoder was updated. As future work, then encoder could be also trained, improving the representations generated at each training epoch.

**Implementation.** As the rest of the architecture, this process has been implemented using the same GPU and parameter configuration, in this case for approximately 6 thousand updates.

### 5.3 Post-processing

Once our model is fully trained the apply an additional step of checkpoint averaging in which the $n$ checkpoints containing the weights of the network are combined using the defaults script provided by *Fairseq*.

In this work, given that the corpus was small we saved checkpoint every epoch of approximately 400 updates and averaged the last 4 checkpoints saved.

Finally, to generate the final submissions, detruecasing and detokenization using the scripts provided by Moses to the English outputs, while IndicNLP detokenization is applied to the Tamil ones.

## 6 Experiments and Results

The motivation for this work was to explore the combination of both positive transfer and monolingual data in a low resource task such as English-Tamil Translation.

To test our hypothesis we trained a bilingual baseline with just the parallel data available for the task and compared its results to an incremental using adaptation to a multilingual NMT system and monolingual fine-tuning to measure the impact of each measure in the final performance. All configurations have the same architecture and number of parameters and have been tested on the same 1275 lines extracted from the *newsdev2020* Tamil-English set.

To introduce some context about the multilingual system, we evaluated its performance using *newstest13* as test set, and the performance English performance ranged from 20.31 BLEU points from the English-German translations direction, to 29.74 for English-French. When English is the target language the results vary from 24.54 for German-English, to 27.75 for Spanish-English. About the impact of positive transfer from Multilingual NMT, Tables 4 and 3 show that both directions benefit from adding Tamil into the MNMT system with improvement of 1.58 and 4.09 BLEU points respectively, approximately a 40% better than the bilingual baseline in both directions.

When looking at the monolingual fine-tuning results, we can observe that the English to Tamil translation direction benefits more (2.65 BLEU) from the technique than the Tamil to English direction (1.02 BLEU). This difference in the performance may be explained by the difference in the training of both decoders. While the Tamil decoder has been trained just with the parallel data for the task, the English decoder was trained with the multilingual NMT system with more data available, which may lead to a more robust model to fine-tuning.

Finally, looking at the checkpoint averaging results, in both directions it leads to a small improvement, less than 0.2 BLEU, showing limited impact in the final results.

| System | BLEU | ΔBLEU |
|---|---|---|
| Baseline | 3.42 | - |
| Multilingual | 5.00 | 1.58 |
| + Mono | 7.65 | 2.65 |
| + Checkpoint Avg | 7.92 | 0.27 |

Table 3: Results measured in BLEU of the English to Tamil Translation direction.

| System | BLEU | ΔBLEU |
|---|---|---|
| Baseline | 6.51 | - |
| Multilingual | 10.6 | 4.09 |
| + Mono | 11.62 | 1.02 |
| + Checkpoint Avg | 11.8 | 0.18 |

Table 4: Results measured in BLEU of the Tamil to English Translation direction.

## 7 Conclusions

In this paper, we described the TALP-UPC participation in the WMT20 news translation shared task for Tamil-English. The motivation of this work was to explore the combination of multilingual transfer from high resource languages and monolingual data applied to low resource NMT. Our experiments showcase the effectiveness of adapting low resource languages pre-trained multilingual systems and how it introduces positive transfer compared to a bilingual baseline system. Also it shows that monolingual data can be successfully introduced into the system and that it can boost the performance of the system. As future work, we could explore the fine-tuning of both encoder and decoder during the monolingual unsupervised fine-tuning in order to help the system produce better synthetic data as the training takes place.

## Acknowledgments

This work is supported in part by the Google Faculty Research Award 2019, the Spanish Ministerio de Ciencia e Innovación, through the postdoctoral senior grant Ramón y Cajal and by the Agencia Estatal de Investigación through the projects EUR2019-103819, PCIN-2017-079 and PID2019-107579RB-I00 / AEI / 10.13039/501100011033

## References

Noe Casas, José A. R. Fonollosa, Carlos Escolano, Christine Basta, and Marta R. Costa-jussà. 2019. The TALP-UPC machine translation systems for WMT19 news translation task: Pivoting techniques for low resource MT. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 155–162, Florence, Italy. Association for Computational Linguistics.

Himanshu Choudhary, Aditya Kumar Pathak, Rajiv Ratan Saha, and Ponnurangam Kumaraguru. 2018. Neural machine translation for English-Tamil. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 770–775, Belgium, Brussels. Association for Computational Linguistics.

Carlos Escolano, Marta R. Costa-jussà, and José A. R. Fonollosa. 2019. From bilingual to multilingual neural machine translation by incremental training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 236–242, Florence, Italy. Association for Computational Linguistics.

Carlos Escolano, Marta R Costa-jussà, José AR Fonollosa, and Mikel Artetxe. 2020. Multilingual machine translation: Closing the gap between shared and language-specific encoder-decoders. *arXiv preprint arXiv:2004.06575*.

Carlos Escolano, Marta R. Costa-Jussà, and José A. R. Fonollosa. From bilingual to multilingual neural-based machine translation by incremental training. *Journal of the Association for Information Science and Technology*, n/a(n/a).

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas.

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, Melbourne, Australia. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL: Demo Papers*, pages 177–180.

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

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. 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, Berlin, Germany. Association for Computational Linguistics.

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.

Changfeng Zhu, Heng Yu, Shanbo Cheng, and Weihua Luo. 2020. Language-aware interlingua for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1650–1655, Online. Association for Computational Linguistics.