The IPN-CIC team system submission for the WMT 2020 similar language task

Luis A. Menéndez-Salazar[1], **Grigori Sidorov**[1], **Marta R. Costa-Jussà**[2]
[1]Centro de Investigacion en Computacion
Instituto Politecnico Nacional, Mexico
[2]TALP Research Center
Universitat Politecnica de Catalunya, Barcelona
menendez.sla@gmail.com, sidorov@cic.ipn.mx, marta.ruiz@upc.edu

Abstract

This paper describes the participation of the NLP research team of the IPN Computer Research center in the WMT 2020 Similar Language Translation Task. We have submitted systems for the Spanish-Portuguese language pair (in both directions). The three submitted systems are based on the Transformer architecture and used fine tuning for domain Adaptation.

1 Introduction

In this paper we describe the Neural Machine Translation(NMT) systems developed by the NLP team of the Computer Research System of the Instituto Politécnico Nacional, México for the similar languages translation shared task of the EMNLP 2020 fifth conference on machine translation (WMT 20).

For this task, we submit systems for both directions of the Spanish-Portuguese language pair, all the submitted systems were based in a transformer architecture with a fine tuning for domain adaptation, the difference of the submitted systems was mainly the kind of tokens used (words and sub-word units) and the initialization of the word embeddings in the systems using either a random initialization or pre-trained word embeddings.

In the past year task, the submissions of the MLLP-UPV team (Baquero-Arnal et al., 2019) showed that the use of fine tuning was proven to be useful in this specific task, although the test corpus of this year had a higher complexity, the use of fine tuning for context adaptation was also beneficial for translation quality.

The paper was organized in the following way. Section 2 describes the architecture used for the systems trained, the section 3 describes the corpora used and the pre-processing of the texts, the section 4 gives a description of the training of the system, section 5 gives a description of the method used for obtaining the translations from the trained system, section 6 shows the results for both the internal evaluation and the evaluation of the task, section 7 is a discussion about the impact of pre-trained word embeddings in the submitted systems and section 8 presents the conclusions.

1.1 Transformer models in NMT

Before transformers most state-of-the-art MT systems relied on recurrent neural networks, with attention mechanisms, but the RNN based architectures although that in theory the information of each token can propagate arbitrary far down in the sequence, due to vanishing gradient in practice when dealing with long sentences or sequences information about the initial tokens can be lost.

As a solution of that problem transformers (Vaswani et al., 2017) are an architecture based in an encoder -decoder approach, but rely mainly in self attention mechanisms.

1.1.1 Encoder

Each encoder layer consists of two components: a self-attention mechanism and a feed-forward neural network. The self-attention mechanism receives a set of encoded representation from the previous layer and weights it in order to generate a set of output encodings. The feed forward network processes each output encoding and passes it to the next encoder and to the decoders.

The first encoder layer uses as arguments positional information and the word embeddings, instead of encodings.

1.1.2 Decoder

Each decoder layer has three main components: A self-attention mechanism, an over the encodings attention mechanism and a feed-forward network. The decoder layer works in a similar way than a encoder one, but the additional attention mecha-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 409–413
Online, November 19–20, 2020.

nism uses the relevant information produced by the encoder layers.

In a similar structure from the first encoder layer, the first decoder layer also receives as inputs positional information and the embeddings of the output sequence. Due to the transformer should not know current or future information in order to predict the next word, the output sequence should be partially hidden during the training of the system.

The last decoder layer is followed by a linear transformation and a softmax layer to produce the probability of the words in the vocabulary.

1.2 Word embeddings initialization

Word embeddings are a solution in Natural Language Processing for the problem of having spare word spaces of high dimensionality that happens with the one-hot vectors representation.

Word embeddings uses a machine learning algorithm in order to learn the relations between words and contexts from big corpora, proposed from (Mikolov et al., 2013)

Inside of neural networks architecture approaches, word embeddings are generally used as a word representation in both source and target languages which are usually random initialized, but it is also possible to use pre-trained word embeddings and updated it in the training time.

Fast text (Bojanowski et al., 2016) is a library used to learn word embeddings, this model aims to create supervised and non-supervised systems to obtain word representation. It is also provided by the project pre-trained embeddings for 294 languages.

In the current paper an approach using fast-text vectors for the initialization of a transformer model is attempted, using the pre trained vectors in both Spanish and Portuguese languages.

2 Architecture of the submission

The main model for the submission consisted in a transformer model that used tokens composed by words and the word embeddings inside the transformer architecture were initialized using pre trained fast text embeddings in both, source and target languages.

For contrastive purposes two additional models were added to the submission , neither of them used a special initialization of the word embeddings and the main distinction between them was the kind of tokens used in the training, the first one used words and the later used sub word units gated by a BPE algorithm.

Corpus	Version	Sentences
JCR	1	1,650,126
Europarl	10	1,801,845
News commentary	15	48,259
Wikititles	2	649,833

Table 1: Training corpora

2.1 Model description

The three models for the submission differs in the following way:

1. **Primary:** Model that was initialized using pre trained fast-text word embeddings, and tokens constituted by words
2. **Contrastive1:** Model that was initialized with random word embeddings. Used tokens formed by words
3. **Contrastive2:** Model that was initialized with random word embeddings. Used tokens formed by BPE sub-word units

Where the primary model was the main model for the submission and the contrastive models serves as baselines.

For the comparative between the three models, BLEU (Papineni et al., 2002) was computed with the `Sacrebleu` (Post, 2018).

2.2 Transformer model

For the transformer model the configuration used consists of a model size of 6 layers, 512 feed-forward size, 8 heads, trained on one GPU with a batch size of 4096 tokens using. We stored a checkpoint every 5000 steps until 200000.

We used Adam optimizer with a β 2 of 0.998 The models were built using Open NMT toolkit (Klein et al., 2017).

3 Corpus description

The training data was made up with the available training data for the task, that is JCR, Europarl, news commentary and wikititles corpora. The provided development set was randomly split in two disjoint sets of the same size, dev1 and dev2 sets.

The data was prepossessed using the following pipeline tokenization, lowercasing and a BPE algorithm learned over the test set.

Model	Before	After
Primary	24.20	32.56
Contrastive1	23.94	30.43
Contrastive2	24.49	30.2

Table 2: BLEU for the models before and after fine-tuning for ES-PT

4 Training

The training for all the systems was carried in a two steps way.

In the first step the model was trained using the training set during 200 thousand steps, storing a checkpoint every 5000 steps. For all the checkpoint generated, a translation of the dev-set was computed and BLEU evaluated against a tokenized and lowercased version of the dev1 set.

Due to the conformation of the training data, that is made mostly of parlament sesions (Europarl) and scientific journals (JCR) and the observation that this domains doesn't appear in the test data there is an assumption of a domain mismatch between training and test data. Due to this mismatch in the second step a fine tunning of the model is conducted.

This fine tuning was trained from the best BLEU scored checkpoint and a retraining was made using the dev1 set up to 3000 steps, storing a checkpoint every 10. For all the stored checkpoints, a translation of the test set was computed and evaluated with BLEU against a tokenized and lowercased version of the dev2 set.

For the generation of the translations for the submission the checkpoint with the best BLEU score in the fine tuning step was used.

5 Translations generation

In order to get the translation for the evaluation the `translate.py` script of Open NMT was used with a beam of size 5 and a length penalization with alpha of 5.

After getting the translations the texts was passed through a recaser trained over the training corpora using the script `recaser.perl` from Moses, after this step a detokenizer was used.

6 Results

The tables 2 and 3 shows the BLEU obtained in the evaluation of the three different models before and after the fine tuning for the ES-PT and PT-ES language pairs respectively.

Model	Before	After
Primary	27.61	34.41
Contrastive1	27.21	34.11
Contrastive2	27.26	34.18

Table 3: BLEU for the models before and after fine-tuning for PT-ES

Model	BLEU	RIBES	TER
Primary	27.08	72.98	55.34
Contrastive1	23.91	71.55	57.55
Contrastive2	23.9	73.73	58.07

Table 4: Official results for submitted ES-PT systems

In this internal evaluation of the models the primary model outperforms the baseline models by 2.7 BLEU points for ES-PT direction and 0.18 points in PT-ES.

6.1 Task results

The evaluation of the task was carried using BLEU, RIBES (Isozaki et al., 2010) and TER (Snover et al., 2006) metrics the main difference between this measure and the internal one was that the internal evaluation used a tokenized lowercased version of the text and the task results used the final version.

The tables 4 and 5 show the results of the submitted systems in the task evaluation for the ES-PT and PT-ES language pairs respectively.

In this evaluation again the primary model outperforms the baseline models by a margin of 3 and 0.4 BLEU points for the ES-PT and PT-ES directions respectively.

7 Impact of the pre trained word embeddings

The pre-trained word embeddings used for the model were filtered in order to include only the words that was present in either the development set or the test set and preprocessed using the script `embeddings_to_torch.py` included in the Open NMT toolset.

The used word embeddings showed an im-

Model	BLEU	RIBES	TER
Primary	28.38	72.24	56.27
Contrastive1	27.98	72.11	56.16
Contrastive2	27.41	75.18	57.28

Table 5: Official results for submitted PT-ES systems

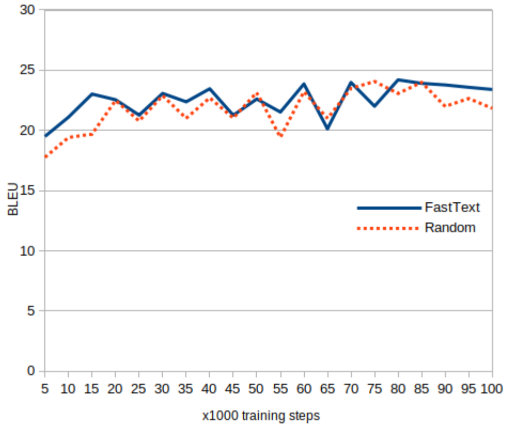

Figure 1: Training for ES - PT

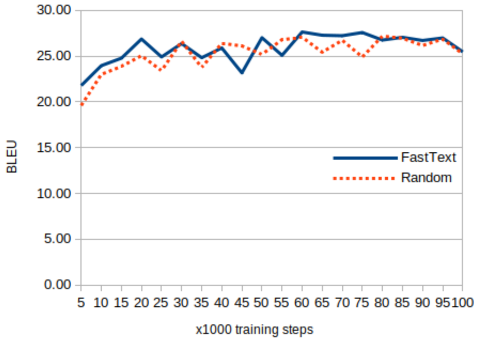

Figure 2: Training for PT - ES

provement in the translation quality (according to BLEU).

Figures 1 and 2 show comparatives for the ES-PT and PT-ES directions respectively for the first 100k training steps for the model 1 and 3, that means comparing both systems trained in words with the unique difference is that system 1 has fast text pre-trained word embeddings.

From the comparative in figures 1 and 2 we can extract that the use of pre-trained word embeddings was beneficial in the beginning of the training, with a difference of around 2 BLEU points in the first 5000 training steps for both ES-PT and PT-ES directions.

Similar results were seen in the fine tuning of the systems, in this case due to the short amounts of epoch for the training of this step a more detailed table is not provided, but in the tables 2 and 3 a difference of 2.7 and 0.3 BLEU points can be seen for the ES-PT and PT-ES directions respectively.

8 Conclusions

The initialization using fast text had a beneficial result in this low resource scenario, but the embeddings used were trained in a general context, is possible that pre-trained the embeddings in the specific context could gather better results.

In this specific experiment both contrastive models had similar results, independently of the kind of tokens used during the training.

Compared with the 2019 edition of the SLT task, this year the test corpus had a different domain, resulting in a lower BLEU score using similar techniques, but also in this year the use of fine tuning improved the translation margin in around 9 points for ES-PT and in almost 7 points for the evaluation using the development set in the primary models.

For the next year submission, the use of word embeddings can be expanded using word embeddings trained in a bilingual context or in a similar domain from the one in the test corpora.

9 Credits

The work was done with support of the Government of Mexico via CONACYT, SNI, CONACYT grant A1-S-47854, and grant SIP 20200797, of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico.

References

Pau Baquero-Arnal, Javier Iranzo-Sánchez, Jorge Civera, and Alfons Juan. 2019. The MLLP-UPV Spanish-Portuguese and Portuguese-Spanish Machine Translation Systems for WMT19 Similar Language Translation Task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 181–186, Florence, Italy. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

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.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.