T4T Solution: WMT21 Similar Language Task for the Spanish-Catalan and Spanish-Portuguese Language Pair

Miguel Canals
miguelknals@gmail.com

Marc Raventós Tato
marcraven@gmail.com

Abstract

This system description describes the participation in the EMNLP 2021 Sixth Conference on MT (WMT21) - Shared Task: Similar translation for the language pairs SPA<>CAT and PTG<>SPA for our T4T solution. The main objective has been to prove that good data with a good standard NMT toolkit, as OpenNMT, is able to provide good results. We have focus in the corpus cleaning (both from the physical and from the statistical side), try to find some alternatives to subword segmentation (syllabic and byte-pair-enconding), and finally use OpenNMT as out-box system with a transformer model. The results have been pretty close to the best ones, if not the best.

1 Introduction

Current available NMT systems have become so complex and resource computing demanding, that the idea behind this project is try to find out if simple logical solutions and standard tools are able to provide good results at least in close languages (according Ethnologue Lexical similarity coef for the language pairs are 0.85 CA<>ES and 0.89 for PT<>ES) (Collin, 2010).

The first thing that come with a little surprise is how we can explain that so similar languages have persistently get so different BLEU (Papineni et al., 2002) scores in previous WMT years, as stated in Table 1 for WMT2020.(Barrault et al., 2020)

ES-CA	CA-ES	ES-PT	PT-ES
86.44	77.08	32.69	33.82

Table 1: Results WMT 2020 for similar languages CA-ES/PT-ES best BLEU score.

We suspect this 50 BLEU score difference is direct result of corpus quality or diversity. CA<>ES corpus provided by the organization uses a very reliable source, the DOG (The official Catalan Government Diary) (approx 40% of words), and even though PT<>ES uses also a similar domain (mainly news and legal), its legal composition (Europarl/JARC) is based in probably a mix of indirect translations. We think one of the best ways to improve a NMT system, is to use the best data you can.

We have have focused in the physical cleaning of the corpus (duplicate strings, unusual sentences, tokenized text in some sources, deal with the UTF-8 universe coding for punctuation, numerical data and the upper/lower casing issue). We have developed a set of python programs for these cleaning tasks and an adhoc tokenizer.

We also have tried also to run some cleaning procedure based in some basic statistical information of the bitext corpus. As there is a quite large source-target, we have scored word probabilities in bitext corpus sentences, and then somehow score sentence probabilities and decide to use or not these sentences. This simple cleaning has indeed increased the score of the model for corpus in-data, but is not so clear if it helps with data out of the corpus.

The last step in data preparation, to deal with the vocabulary size issue, has been the subword segmentation. We have used python standard tools for syllabic segmentation with good results (corpus data has achieved best score than BPE (byte-pair-enconding), but again, with data outside of the corpus, BPE (Sennrich et al., 2015) has proven better. At the end, we have used Google SentencePiece (Kudo and Richardson, 2018) BPE implementation.

After that, we have used what we think a proven toolkit for NMT, OpenNMT (Klein et al., 2017), out of the box, without any modification, using its web publish options for the Transformer model. In the last step we have used the inverse python programs in order to generate the final version of the test source translated file.

We have focused the system from a practical engi-

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 279–283
November 10–11, 2021.

neering point of view. The whole project has been based in currently available local only mid size system, with consumer grade multi-gpu environment. Following this fast approach, and due the nature of the main focus on the corpus, we have used simple models that OpenNMT provides (2-layer LSTM with 500 hidden units on both the encoder and decoder) in order to choose several options and parameters used later for the final transformer model, as this last model is close to the limit a midsize system can provide.

2 Cleaning the corpus

In our approach we have joined all data sources (all monolingual sources and the matching language for the bilingual text corpus) in order to create a mono corpus, and the bilingual text corpus. For the model training we have used the dev data provided by the organization. The typical size for this file in OpenNMT is around 2000 lines, so we have used data from the bilingual text corpus in order to reach this typical size.

2.1 Physical cleaning steps for the bilingual corpus

These are the direct "physical" tasks in order to prepare a corpus with what they look "standard" sentences.

- Removal of duplicated sentences.
- As many strings are already tokenized we have detokenize all the corpus with the Moses (Koehn et al., 2007) detokenizer, as our custom tokenizer works with untokenized text. (Some punctuation is changed, but indeed fixes many punctuation format errors as coma not correctly joined to the words).
- Perform physical cleaning. These are some steps based on the manual inspection of the corpus in order to remove noise sentences or fix others. For instance, remove all left chars until an alphabetic char is found, remove some keywords leftovers, sentences should have at least a spell correct word (Németh, 2010), remove any text between parenthesis or remove duplicates.
- Using python nltk (Bird et al., 2009) package we have removed all sentences that probably are made up of more than one sentence.

2.2 Statistical cleaning for the bilingual corpus

Using the bitext corpus we have created source/target dictionary and all instances of where source word and target word appears in the same sentence pair. Using simple rules we can try to score the probability of the source word given a target word, and somehow score words and sentences. Then create a list and remove the ones with worst scores. Most of this cleaning is based in heuristic parameters.

The clean is indeed effective as for instance, score for corpus data for PT->ES using this cleaning can raise from 49,38 BLEU score to 67.27, but these gains are not matched when we have used test data outside the corpus. We suspect, this cleaning creates an ideal statistical data set that cannot explain "real" data outside the corpus. So we have used this feature in a moderate way, removing 15% of the low matching sentences. Many of these sentences are indeed removed for a good reason, but many times too are not, because translators many times do not follow a statistical behavior.

We suspect this is an open field. This "cleaning" is close related to word alignment, so probably it would have been wiser use some GIZA++(Och and Ney, 2003) or fastalign word based solution.

3 Tokenization

One of the big issues to deal with real data, is the tokenization. After reviewing several available tools, we ended creating a python custom tokenizer that has the following features.

- It uses a list of split chars (comma, dot, hyphen, ...). The number of these chars that are not alphabetic can be quite large, and is a source of many problems. This list of split chars is generated by the tokenizer itself in a first scanning phase.
- Numbers are replaced by variables (as ((n0)), ((n1)). These numbers are kept in an independent file in order to be used if detokenization is required. This will avoid the use of numbers, another big source of undesired vocabulary.
- Casing is indicated with special tags before the upper word in to ways, ((up)) for first uppercase only first letter words, or ((aup)) for

all uppercase letter words. This avoids most of casing issues and allow us to work with a full lowercase input in the neuronal toolkit.

- Tokenizer also keeps track of the spaces for words and split chars.

These features provide a robust tokenization <> detokenization reversibility.

Sentence example:

> Vista la Directiva 91/494/CEE del Consejo, de 26 de junio de 1991, sobre ...

is tokenized as:

> ((up)) vista la ((up)) directiva ((n0)) @@/@@ ((n1)) @@/@@ ((aup)) cee del ((up)) consejo @@, de ((n2)) de junio de ((n3)) @@, sobre

4 Word segmentation: BPE and syllabic

Word segmentation is further step in order to to reduce neuronal network vocabulary.
We have followed two approaches, the well know BPE subword segmentation, but also an uncommon one, a syllabic segmentation. We have used again a known python tool (https://pyphen.org/) to split words in syllables.
Results are quite interesting as they have been quite consistent. Using a syllabic segmentation:

- BLEU scores for corpus test data in all NTM models (LSTM or Transformer) have been better.
- BLEU scores for external data have been worst.

So the promising syllabic segmentation, has not responded so well with data outside de corpus. Due this, BPE has been chosen for final models.

5 Evaluation

After testing in more simple neuronal network models (LSTM) the final setup has consisted of a corpus cleaned (in the physical sense, and also in an statistical sense removing approx 15% of sentence corpus sentences with the highest perplexity). This clean corpus has been detokenized with our adhoc tokenizer (that lowercase the corpus, replaces numbers by variables, and handles punctuation and upper/lower casing).

After this cleaning, the number of words for ES<>PT has been around 2M lines (55.3M words) and ES<>CA around 9.5M lines (176M words).

Then we have used 16000 terms SentencePiece BPE vocabulary on this detokenized corpus in order to reduce vocabulary. We have removed sentences with more than approx 170 tokens for the sentences the neuronal network has ingested (This length has kept the model below the memory limit of each one of the GPU cards).

We have set the model configuration using the published Transformer(Uszkoreit, 2017) model in the OpenNMT site (https://opennmt.net/OpenNMT-py/FAQ.html#how-do-i-use-the-transformer-model). According OpenNMT documentation, this setup mimic the Google (Vaswani et al., 2017) setup that replicate its WMT results.

We have tested our models against test data form our corpus (23_SP_TRANSF_Statclean_3,5 for PT<>ES and 25_SP_CAES_2_TRANSF_Statclean_3.5 for CA<>ES) and also from the test data from WMT2020 (test20 for PT<>ES and test20.v2 CA<>ES).

In Figure 1 we can see the results for the PT<>ES for both test sets and both directions. The transformer model converges really fast after 30-40K steps (as the size of the corpus is not very large). We have used the best score (PT->ES BLEU score 55.96 at 55K steps and ES->PT BLEU score 54.68 at 60K steps) for the final evaluation.

In Figure 2 we can see the results for the CA<>ES for both test sets and both directions. We have used the best score (CA->ES BLEU score 84.34 at 70K steps and ES->PT BLEU score 83.77 at 85 steps) for the final.

6 Results

In Table 2 we can compare the best score of each one of the 3 teams that have submitted results for this WMT 2021 task (http://www.statmt.org/wmt21/similar.html).

7 Conclusions

We think we have accomplish the objective to achieve good results with good data and out of box toolkit as OpenNMT.
It has proven more difficult than expected find recipes to improve the corpus quality beyond the physical cleaning. What we have found suggest (without prove) that:

- Cleaning the corpus trying to remove sentences with low translation probability to be correct looks to us that can improve the corpus for sure, but is not so clear will happen the same for data outside the corpus. The idea of find correct paired translated sentences in the bitext, is a translation/alignment problem by itself, and probably the simple statistical system we have used has much room to improve.
- Syllabic word sub segmentation can improve greately the corpus quality, but has not improved the score with data outside the corpus. The reason is unknown.

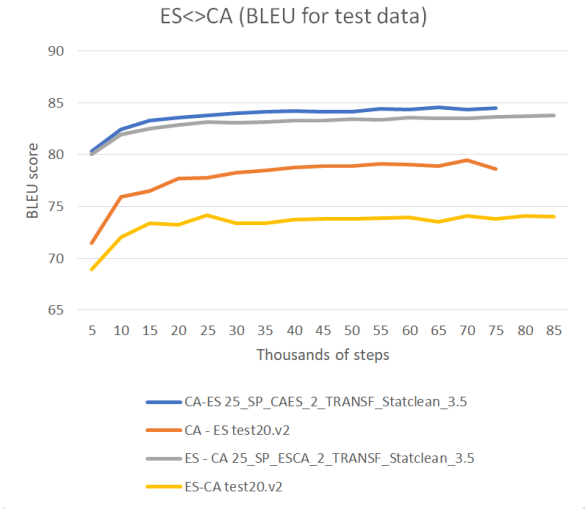

Figure 1: Results PT<>ES BLEU score for test data from the corpus and external to the corpus

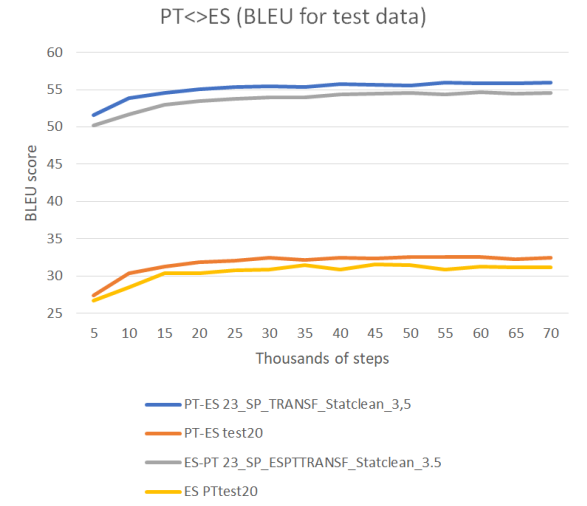

Figure 2: Results CA<>ES BLEU score for test data from the corpus and external to the corpus

	BLEU		RIBES		TER	
	Best score	T4T	Best score	T4T	Best score	T4T
PT-ES	47.71	46.29	87.11	87.04	39.21	40.12
ES-PT	**40.74**	**40.74**	**85.69**	**85.69**	**43.34**	**43.34**
CA-ES	82.79	77.93	96.98	96.04	10.92	16.5
ES-CA	79.69	78.60	**96.24**	**96.24**	14.63	16.13

Table 2: Results for the bests system and T4T

References

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Richard Oliver Collin. 2010. Ethnologue. *Ethnopolitics*, 9(3-4):425–432.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Philipp Koehn, Marcello Federico, Wade Shen, Nicola Bertoldi, Ondrej Bojar, Chris Callison-Burch, Brooke Cowan, Chris Dyer, Hieu Hoang, Richard Zens, et al. 2007. Open source toolkit for statistical machine translation: Factored translation models and confusion network decoding. In *CLSP Summer Workshop Final Report WS-2006, Johns Hopkins University*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

László Németh. 2010. Hunspell. *Dostupno na: http://hunspell. sourceforge. net/[01.10. 2013]*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

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. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Jakob Uszkoreit. 2017. Transformer: A novel neural network architecture for language understanding. *Google AI Blog*, 31.

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.