OSTI: An Open-Source Translation-memory Instrument

Anonymous COLING submission

Abstract

We present OSTI: a free open-source tool to process and visualize a pair of bilingual documents (original and translation) into automatically labeled sentence pairs. This can be used by translation professionals as a human-accessible quality evaluation tool, as a pre-processing step for human annotation as well as an intermediate step to populate a Translation Memory.

1 Introduction

The development of Computer-Assisted Translation (CAT) tools started gaining popularity in the mid 1980s. Since then, the elaboration of new and more sophisticated CAT tools has not ceased to increase. Given that training data is essential to the development of high-end automatic models, there has been great academic and (sometimes) corporative efforts to make large and clean Translation Memories (TM) and translation data sets (bilingual and monolingual dictionaries, terminologies, etc.) publicly available. Works such as (Koehn, 2005; Tiedemann, 2012; Steinberger et al., 2013) have greatly facilitated the elaboration of pioneering CAT tools *freely* available today.

Nevertheless, a problem arises when users have their own proprietary data and want to use it instead of the general and publicly available one. Multiple companies offer to tailor-made an exclusive TM using proprietary data but this service can be a quality black-box, unaffordable to the more humble translators/translation companies or apprehensive to companies working with sensible data.

Having this in mind, we designed a simple yet (hopefully) useful open-source tool which takes as input a pair of supposedly parallel documents in English and French[1] which are segmented and aligned into units whose quality is automatically inspected, labeled and presented as an easy to consume HTML (an example of which is reported in Figure 1) and can subsequently be distilled into a TMX format.

Our tool can be used as a human-accessible quality evaluation tool, as a pre-processing step for human annotation, as well as an intermediate step to populate a Translation Memory.

2 System overview

We conceived OSTI in a modular fashion, trying to make the integration of new components, tools or language pairs as easy as possible. Its has been currently tested on the French-English language pair (our use case), but it should apply with minor or no adaptation to other language pairs.

The overall pipeline is depicted in Figure 2 and embeds 4 modules: a) we first segment each text into sentences, that b) we align at the sentence level; c) the sentence pairs (SPs) are then classified into good or bad SPs; d) the SPs are further labelled into 6 classes for the sake of human readability on an HTML interface that allows to export the automatically/manually selected SPs into a TMX file.

2.1 Sentence Segmenter

Even though the task of sentence segmentation is not a very enticing one, it is crucial to have a clean sentence segmentation to start with in order to reduce the subsequent tasks. We use the NLTK sentence tokenizer as a default, although we also considered Spacy (Explosion, 2017) and the `Mediacloud`

[1] We targeted English and French languages, but arguably many components could be used for other language pairs. To keep it simple, we also assume that documents are converted into text prior to using OSTI.

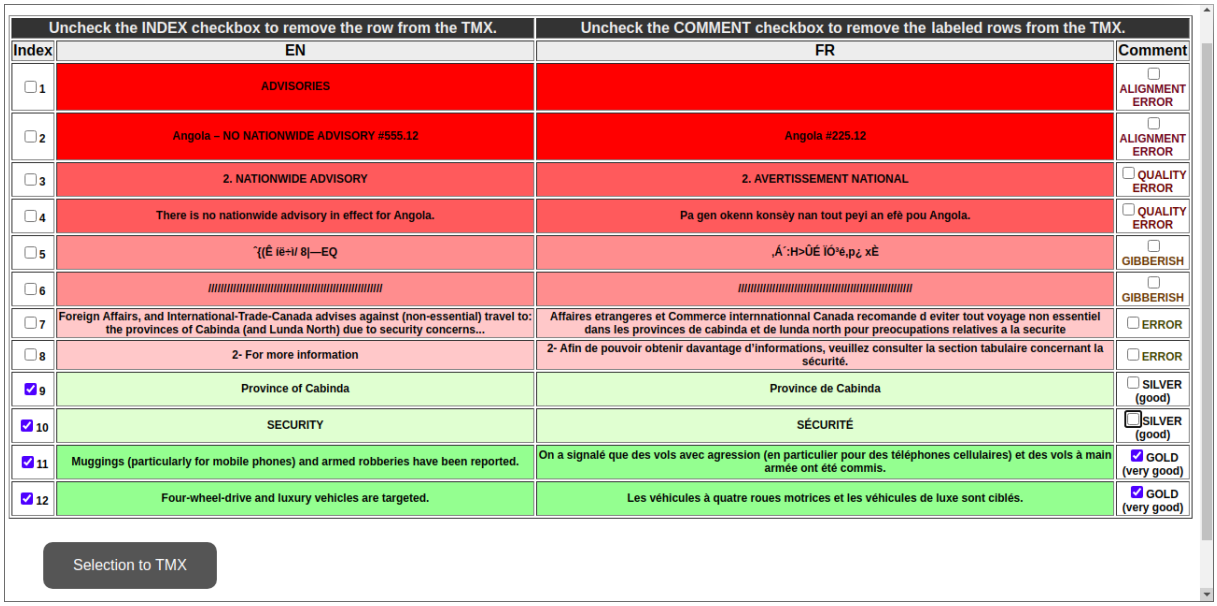

Figure 1: Screenshot of OSTI's HTML visualization. Each sentence pair is presented in different colours according to their gauged quality (see the text for more). As a default, `Gold` and `Silver` sentence pairs are selected for conversion to TMX, but is is up to the user to change this selection by removing/adding individual sentence pairs (*Index* checkbox on the left) or by removing/adding all sentence pairs having the same label (*Comment* checkbox on the right).

Sentence Splitter[2]. After a small empirical analysis, NLTK showed to output the best out-of-the-box results for our specific language pair, we therefore selected it as our default segmenter.

2.2 Bilingual Sentence Aligner

We benchmarked two very different tools that have both shown to be accurate and robust: YASA (Lamraoui and Langlais, 2013) and VECALIGN (Thompson and Koehn, 2019). The former system is very similar to `HunAlign` (Varga et al., 2007), that is, a sentence-length score (Gale and Church, 1993) enhanced with a cognate-based one (Simard et al., 1992), and has been reported faster and more accurate than more elaborated systems such as BMA (Moore, 2002). The VECALIGN system uses a more innovative scoring function specially made to work with sentence embedding vectors. Although this system is said to accommodate embeddings from various toolkits, we used the recommended multilingual Language-Agnostic SEntence Representations (LASER) from (Artetxe and Schwenk, 2019).

[2]https://github.com/berkmancenter/mediacloud-sentence-splitter (based on the implementation by (Koehn, 2005))

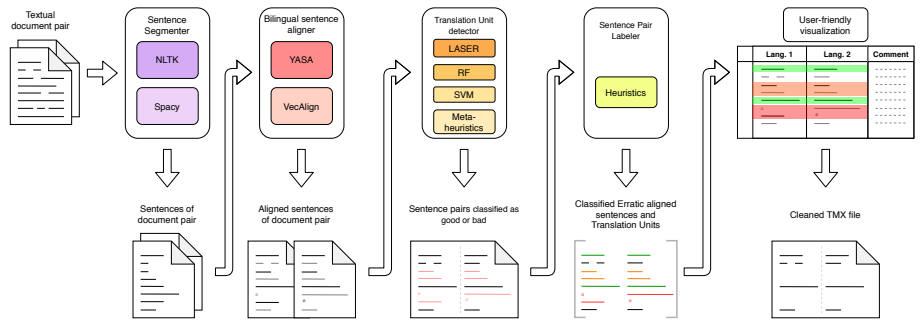

Figure 2: Pipeline of OSTI from a document pair to the visualization tool. Darker components are the default ones.

Since our target language pair is French-English, we compared both aligners using the BAF corpus benchmark (Simard, 1998) which contains 11 document pairs of 4 different genres (Literary, Institutional, Scientific, and Technical) segmented into approximately 25k sentences (in each language) aligned and manually checked. To evaluate the aligners, we used the benchmark's metrics of precision, recall and F1 at the alignment pair level and at the sentence level. See (Langlais et al., 1998) for a description of those metrics.

		Literacy		Institutional		Scientific		Technical		**All doc.**	
		YASA	VECA	YASA	VECA	YASA	VECA	YASA	VECA	YASA	VECA
align.	Prec.	0.59	0.61	0.94	0.95	0.86	0.86	0.85	0.86	0.86	**0.87**
	Rec.	0.74	0.71	0.95	0.95	0.93	0.91	0.96	0.95	**0.92**	0.91
	F_1	0.65	0.65	0.94	0.95	0.89	0.88	0.90	0.90	**0.89**	**0.89**
sent.	Prec.	0.88	0.87	0.98	0.98	0.99	0.98	0.99	0.98	**0.98**	0.97
	Rec.	0.79	0.84	0.94	0.95	0.86	0.86	0.06	0.06	0.81	**0.82**
	F_1	0.83	0.86	0.96	0.96	0.92	0.91	0.11	0.11	**0.85**	**0.85**

Table 1: Alignment and sentence Precision, Recall and F_1 scores of YASA and VECALIGN as a function of text genre on the BAF benchmark. The last columns aggregates those measures by averaging over all document pairs, regardless of genre.

Results are reported in Table 1. There are two observations that can be made. First, both systems deliver very similar performances. Second, the performance varies substantially depending on the text genre, the worst setting is when aligning literary texts[3]. We refer the reader to (Xu et al., 2015) for extensive comparisons of alignment techniques on literacy texts. Because both systems perform on par, we selected YASA as our default sentence aligner since is it much lighter than VECALIGN in terms of hardware (CPU versus GPU).[4]

2.3 Translation Unit Detector

This component consists in deciding whether a given sentence pair contains a problem or not. By itself, the sentence pairs that are classified as good are elected Translation Units (TU) and can be saved into a TMX format for further consumption. In order to do so, we implemented 5 families of classifiers whose details are reported elsewhere: a) an heuristic-based approach (designated as META-HEURISTICS in Figure 2) involving 13 heuristics further introduced in Sub-section 2.4; b) feature-based classifiers (62 features); c) a re-implementation of the BI-LSTM approach described in (Grégoire and Langlais, 2018); d) a cleaning method devised internally on top of the LASER model developed and pre-trained by (Artetxe and Schwenk, 2018); and e) the unsupervised TM cleaning tool developed by (Jalili Sabet et al., 2016) that relies on heuristics, IBM-like features and unsupervised word-embeddings.

We compared those classifiers in terms of accuracy on a manually annotated proprietary corpus of 2021 sentence pairs and found LASER (0.84) to outperform biLSTM (0.79) and largely surpassing other ones: feature-based classifier (0.63), TMOP (0.60) and heuristics (0.42). Considering that LASER is unsupervised, and relatively fast to use, we selected it as our default detector.

2.4 Sentence Pair Labeler

There are many reasons why sentence pairs can be detected erroneous, including bad sentence alignment (often caused by sentence segmentation issues), errors in translations (calques, false friends, etc.), as well as encoding issues (which happen in complex organisations due to numerous format manipulations). All of these are considered errors but not all are equally important to the translation professional. Therefore, we take an extra step into further labelling sentences pairs into 6 labels we sketch in the sequel. We do

[3]There is one pair of texts in BAF belonging to this category which is a novel of Jules Vernes where the English version is abridged, which confuses the dynamic programming optimization driving each approach.

[4]VECALIGN accommodates CPU computations but delivering much slower time responses.

this by taking advantage of the 13 heuristics embedded in the META-HEURISTIC component of the TU detector. Each heuristic was conceived to detect particular types of errors.

`Aligt` qualifies a major mismatch, such as numerical entities issues (row 2 in Figure 1), sentence length issues (row 1 in Figure 1), etc. In total, 8 concerned heuristics.

`Quality` characterizes mistakes identified by 5 heuristics: misspelling issues (row 4 in Figure 1), false-friends, sentences that are part of table of content or indexes that are often misaligned (row 3 in Figure 1), and non-translated target sentences (which happens when documents are partly translated).

`Gibberish` qualifies sentence pairs that contain mainly gibberish (row 6 in Figure 1), sometimes due to encoding issues (row 5 in Figure 1).

`Error` it is not always reasonable to attribute a problem to a specific cause, and when we fail, we use this label. Row 7 (punctuation mismatch and misspelling issues) and row 8 (table of content detection and length mismatch) in Figure 1 are such examples where both `Aligt` and `Quality` compete.

`Silver` sometimes, the TU detector classifies a sentence pair as good but at least one heuristic indicates the presence of a problem (row 9 and 10 in Figure 1).

`Gold` qualifies SPs that are classified good by the detector, and for which no heuristic indicates any problem (row 11 and 12 in Figure 1).

2.5 Technical details

We measured the time response of OSTI on 2 document pairs containing around 900 sentences each (1 800 sentences/160k characters in total). We used a computer with 30Gb of RAM, a 12-core CPU and a GeForce GT 1030 GPU (used by LASER and VECALIGN). On average, sentence segmentation takes less than 0.5 seconds. Sentence alignment with YASA takes 13 seconds (CPU) and 17 seconds with VECALIGN (GPU). Concerning the TU detector, the fastest is the heuristic-based approach (112 seconds, CPU), followed by LASER (117 seconds, GPU) and, lastly, the 2 feature based classifiers (SVM: 131 seconds, RF: 239 seconds, both using a single CPU). Finally, to save time, the sentence pair labeler is run concurrently to the TU detector.

Our Github project[5] requires the installation of a small amount of modules (such as NLTK, NUMPY, FAISS or PYTORCH) that are specified in our requirements file and that are easily installed using the standard `pip` package manager or by executing the setup file. It also includes all the in-house algorithms/implementations written in Python as well as the pre-trained models mentioned in Section 2.3. Finally, it also contains tools, such as YASA and LASER, that are compiled to work on Debian-based Linux distributions. We believe this allows sufficient freedom to the user for the time being.

3 Conclusion

We presented OSTI, an open-source tool which detects good from bad sentence pairs in a French-English pair of (supposedly) parallel documents. These are then further labelled into a set of 6 labels that can be inspected with a simple Web browser and easily transformed into a TMX file. OSTI can be used as a human-accessible quality evaluation tool, as a pre-processing step for human annotation, as well as an intermediate step to populate a Translation Memory. We are aware of proprietary solutions, but there is a striking absence of open-source and peer reviewed such systems.

Currently, we targeted the English-French langage pair, which we plan to revisit. We do not anticipate much difficulties since most components involved in OSTI are arguably language agnostic. Also, we distribute a simple batch pipeline, while for professional use, a client-server application should be of better help. In benchmarking embedded components, we were surprised by the fact that a simple sentence aligner was performing on par with a much more recently engineered one. We plan to revisit this benchmarking on other language pairs and conditions.

[5] Anonymized URL.

References

Mikel Artetxe and Holger Schwenk. 2018. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

AI Explosion. 2017. spacy-industrial-strength natural language processing in python. *URL: https://spacy. io*.

William A. Gale and Kenneth W. Church. 1993. A program for aligning sentences in bilingual corpora. *Comput. Linguist.*, 19(1):75–102, March.

Francis Grégoire and Philippe Langlais. 2018. Extracting parallel sentences with bidirectional recurrent neural networks to improve machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1442–1453, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Masoud Jalili Sabet, Matteo Negri, Marco Turchi, José GC de Souza, and Marcello Federico. 2016. Tmop: a tool for unsupervised translation memory cleaning. pages 49–54.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Fethi Lamraoui and Philippe Langlais. 2013. Yet another fast, robust and open source sentence aligner. time to reconsider sentence alignment. *XIV Machine Translation Summit*.

Philippe Langlais, Michel Simard, and Jean Vronis. 1998. Methods and Practical Issues in Evaluating Alignment Techniques. In *36th Annual Meeting of the Association for Computational Linguistics (ACL) and 17th International Conference on Computational Linguistic (COLING)*, Montreal, Canada, Aug.

Robert C. Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users*, AMTA '02, pages 135–144.

Michel Simard, George Foster, and Pierre Isabelle. 1992. Using cognates to align sentences in bilingual corpora. In *in Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 67–81.

Michel Simard. 1998. The baf: a corpus of english-french bitext. In *First International Conference on Language Resources and Evaluation*, volume 1, pages 489–494.

Ralf Steinberger, Andreas Eisele, Szymon Klocek, Spyridon Pilos, and Patrick Schlüter. 2013. Dgt-tm: A freely available translation memory in 22 languages. *arXiv preprint arXiv:1309.5226*.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1342–1348, Hong Kong, China, November. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218.

Dániel Varga, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2007. Parallel corpora for medium density languages. *Amsterdam Studies In The Theory And History Of Linguistic Science Series 4*, 292:247.

Yong Can Xu, Aurélien Max, and Franois Yvon. 2015. Sentence alignment for literary texts. *Linguistic Issues in Language Technology*, 12.