Dodging the Data Bottleneck: Automatic Subtitling with Automatically Segmented ST Corpora

Anonymous ACL submission

Abstract

Speech translation for subtitling (SubST) is the task of automatically translating speech data into well-formed subtitles by inserting subtitle breaks compliant to specific displaying guidelines. Similar to speech translation (ST), model training requires parallel data comprising audio inputs paired with their textual translations. In SubST, however, the text has to be also annotated with subtitle breaks. So far, this requirement has represented a bottleneck for system development, as confirmed by the dearth of publicly available SubST corpora. To fill this gap, we propose a method to convert existing ST corpora into SubST resources without human intervention. We build a segmenter model that automatically segments texts into proper subtitles by exploiting audio and text in a multimodal fashion, achieving high segmentation quality in zero-shot conditions. Comparative experiments with SubST systems respectively trained on manual and automatic segmentations result in similar performance, showing the effectiveness of our approach.

1 Introduction

Massive amounts of audiovisual content are available online, and this abundance is accelerating with the spread of online communication during the COVID-19 pandemic. The increased production of pre-recorded lectures, presentations, tutorials and other audiovisual products raises an unprecedented demand for subtitles in order to facilitate comprehension and inclusion of people without access to the source language speech. To keep up with such a demand, automatic solutions are seen as a useful support to the limited human workforce of trained professional subtitlers available worldwide (Tardel, 2020). Attempts to automatise subtitling have focused on Machine Translation for translating human- or automatically-generated source language subtitles (Volk et al., 2010; Etchegoyhen et al., 2014; Matusov et al., 2019; Koponen et al., 2020). Recently, direct ST systems (Bérard et al., 2016; Weiss et al., 2017) have been shown to achieve high performance while generating the translation in the target language without intermediate transcription steps. For automatic subtitling, Karakanta et al. (2020a) suggested that, by directly generating target language subtitles from the audio (i.e. predicting subtitle breaks together with the translation), the model can improve subtitle segmentation by exploiting additional information like pauses and prosody. However, the scarcity of SubST corpora makes it hard to build competitive systems for automatic subtitling, especially if no corpus is available for specific languages/domains.

One solution to the SubST data bottleneck could be leveraging ST corpora by inserting subtitle breaks on their target side. Automatic segmentation of a text into subtitles is normally implemented with rule-based approaches and heuristics, e.g. a break is inserted before a certain length limit is reached. More involved algorithms (SVM, CRF, seq2seq) predict breaks using a segmenter model trained on subtitling data for a particular language (Álvarez et al., 2016, 2017; Karakanta et al., 2020c). Still, the performance of these models relies on high-quality segmentation annotations for each language, which web-crawled subtitling corpora like OpenSubtitles (Lison et al., 2018) rarely contain.

In this work, we address the scarcity of SubST corpora by developing a multimodal segmenter[1] able to automatically annotate existing ST corpora with subtitle breaks in a zero-shot fashion. Specifically, our segmenter exploits, for the first time in this scenario, the source language audio (here: En) and segmented target text already available in a few languages (here: De, En, Fr, It). Its key strength is the ability to segment not only target languages for which high-quality segmented data is available but also unseen languages having some degree of similarity with those covered by the original ST

[1]The code and the model will be released upon acceptance.

resource(s). This opens up the possibility to automatically obtain synthetic SubST training data for previously not available languages. Along this direction, our zero-shot segmentation results on two unseen languages (Es, Nl) show that training a SubST system on automatically-segmented data leads to comparable performance compared to using a gold, manually-segmented corpus.

2 Methodology

Our method to leverage ST corpora for SubST can be summarized as follows: *i)* we train different segmenters on available human-segmented subtitling data in order to select the best performing one; *ii)* we run the selected segmenter in a *zero-shot* fashion (i.e. without fine-tuning or adaptation) to insert subtitle breaks in unsegmented text data of *unseen* languages; *iii)* then, the automatically annotated texts are paired with the corresponding audio to obtain a synthetic parallel SubST corpus; *iv)* finally, a SubST model is trained on the synthetic corpus.

We test our method on two language pairs, by comparing the results of SubST models trained on synthetic data with those of models with identical architecture but trained on original gold data.

2.1 Segmenter

We adopt the general segmentation approach of (Karakanta et al., 2020b) where a **textual segmenter** takes unsegmented text as input and inserts subtitle breaks. Since subtitling constraints are the same across several languages, our first extension to this approach is to learn segmentation multilingually. To this aim (see Appendix A), we combine samples from multiple languages in the same training step (Ott et al., 2018) and add a prefix language token to the target text (Inaguma et al., 2019). As in MT (Ha et al.), multilingual training has been shown to enhance ST performance (Wang et al., 2020) while allowing for maintaining only one model for multiple languages. Since in preliminary experiments (see Appendix B) we found multilingual training to be more effective than training a model for each language, we opted for adopting multilingual training for all our segmenters.

Our second extension is multimodal training. Since speech phenomena, such as pauses and silences, can strongly influence the structure of the subtitles (Carroll and Ivarsson, 1998), we expect that information from the speech modality could improve segmentation. To explore this hypothesis, we extend the multilingual segmenter with a multimodal architecture (Sulubacak et al., 2020), which receives input from different modalities: in our case, audio and text.[2] Our **multimodal segmenter** is built using an architecture with two encoders: one for the text (with the same structure as the textual segmenter) and one for the audio. We combine the encoder states obtained by the two encoders using parallel cross-attention (Bawden et al., 2018),[3] as it proved to be effective both in speech and machine translation (Kim et al., 2019). Parallel attention is computed by attending at the same intermediate representation (the decoder self-attention); then, the audio encoder cross-attention and the text encoder cross-attention are summed together and fed to the feed-forward layer.

2.2 Data and Evaluation

Data. To train our textual and multimodal segmenters, we use En→{De/Fr/It} sections of MuST-Cinema (Karakanta et al., 2020b), the only publicly available SubST dataset. To test the segmenters in zero-shot conditions (Section 3) and train our SubST models (Section 4), we select two target languages also contained in MuST-Cinema:[4] Dutch (an SOV – Subject-Verb-Object – language) and Spanish (SVO). Using the corpus notation, subtitle breaks are defined as: *block break* `<eob>`, which marks the end of the current subtitle displayed on screen, and *line break* `<eol>`, which splits consecutive lines inside the same block.

Baselines. We compare the performance of the segmenters with two baselines. One is a rule-based method (*Count Chars*) where a break is inserted before a 42-character limit. This is the simplest method to always produce length-conforming subtitles and serves as a lower bound for segmentation performance. Our second baseline (*Supervised*) is a neural textual segmenter trained on OpenSubtitles, the largest collection of publicly available textual subtitling data, for the respective language (Es, Nl). Although OpenSubtitles is available for a variety of languages, it has some limitations: it does not

[2] Images and videos are normally protected by copyright and thus not publicly available. In case suitable training data and benchmarks will become publicly available, we leave to future work their use to improve the segmenter performance.

[3] We also tested sequential cross-attention (Zhang et al., 2018) but we do not report its results since they are slightly worse compared to the parallel one.

[4] Though present in MuST-Cinema, Es and Nl data are only used for testing purposes so as to simulate the zero-shot conditions required to select the best segmenter and evaluate our SubST systems.

contain audio, the subtitle and segmentation quality varies since subtitles are often machine-translated or created by non-professionals, and line breaks were lost when pre-processing the subtitles to create the corpus. These limitations may have a detrimental effect on the quality of segmenters trained on this data (Karakanta et al., 2019). Complete details on data, baselines and experimental settings are presented in Appendix A.

Evaluation. To evaluate both the quality of the SubST output and the accuracy of our segmenters, we resort to reference-based evaluation. Since to date there is no single metric for the plausibility of breaks positioning, we use BLEU (Post, 2018)[5] to measure the similarity (n-gram overlap) of the generated segmentation (*pred*) with the reference (*ref*).[6] We compute it both *with* the inserted breaks (BLEU) and *without* them (BLEU_{nb}) in order to spot any undesired changes made to the original text. To ensure that the system does not over- or under-generate subtitle breaks, we additionally report *Break coverage* computed as follows:

$$Coverage(\%) = \left(\frac{\#\texttt{<break>}_{pred}}{\#\texttt{<break>}_{ref}} \cdot 100\right) - 100$$

where <break> corresponds to either <eol> or <eob>. EOL and EOB coverage obtains negative values when the segmenter inserts less breaks than required or positive values when it inserts more. Lastly, we use *length conformity* (or characters per line – CPL), corresponding to the percentage of subtitles not exceeding the allowed maximum length of 42 CPL, as per TED guidelines.[7]

3 Zero-shot segmentation

Towards building a SubST model for unseen languages (Es and Nl), we first select the best segmenter for generating synthetic En→Es/Nl data.

As shown in Table 1, all the models that receive only text as input (*Count Chars*, *Supervised* and *Textual*), result in low segmentation performance. The zero-shot *Textual* segmenter achieves the lowest segmentation quality, as shown by a BLEU score of 52.6 for Nl and 54.5 for Es. In this respect, the *Count Chars* and *Supervised* baselines perform slightly better, but this difference comes from less changes to the actual text, as shown by the higher BLEU_{nb} scores. Moreover, both *Supervised* and *Textual* generate subtitles conforming to the CPL constraint in only 70% of the cases, despite having received only length-conforming subtitles as training data. The negative values of EOL and EOB coverage show that all textual methods under-generate subtitle breaks. Despite being trained on subtitling data for the particular language, the low performance of *Supervised* can be attributed to the different domain compared to the MuST-Cinema test set. For example, MuST-Cinema mainly contains long sentences with multiple breaks, while in OpenSubtitles we rarely come across sentences with more than three breaks. From these results we can conclude that zero-shot segmentation does not perform satisfactorily with textual input only.

		Dutch			
Segmenter	BLEU	BLEU_{nb}	CPL	EOL	EOB
Count Chars	61.9	**100**	**100%**	**-21.2%**	-7.1%
Supervised	60.4	89.5	71.2%	-31.4%	-51.3%
Textual	52.6	61.3	77.8%	-23.4%	-9.9%
Multimodal	**80.1**	99.9	91.4%	-27.2%	**+0.4%**
		Spanish			
Segmenter	BLEU	BLEU_{nb}	CPL	EOL	EOB
Count Chars	62.5	**100**	**100%**	-24.6%	**-4.4%**
Supervised	62.1	92.6	71.2%	-32.3%	-45.4%
Textual	54.5	69.6	70.1%	-47.6%	-19.3%
Multimodal	**80.6**	99.6	91.8%	**-22.4%**	+4.7%

Table 1: Segmentation results on *unseen* languages.

In comparison, the *Multimodal* segmenter performs significantly better. It reaches an absolute gain of 27.5 BLEU for Nl and 26.1 BLEU for Es compared to *Textual*. Moreover, contrary to *Textual* and *Supervised*, the *Multimodal* model learnt to perfectly copy the text, as shown by the high BLEU_{nb} scores (up to 99.9 on Nl), close to the maximum score of a method – *Count Chars* – that by design does not change the original text. The CPL results are in agreement with BLEU: for both languages, the *Multimodal* model respects the length constraint in more than 91% of the subtitles. Strikingly, even if the two target languages were never seen by the model, these results are similar to those obtained on seen languages (see Appendix B). Unlike the rest of the models, *Multimodal* is the only model that does not under-generate <eob>. This is in line with the results of Karakanta et al. (2020a), who showed that exploiting the audio in ST is beneficial for inserting subtitle breaks (<eob>, for instance, typically corresponds to longer speech pauses). The results are more discordant for the

[5]BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.5.1

[6]We do not use F1, a standard accuracy metric used in segmentation tasks, since it cannot be computed when the output text is different than the reference.

[7]https://www.ted.com/participate/translate/subtitling-tips

EOL Coverage. On Es, *Multimodal* shows a lower tendency to under-generate, while on Nl both models fail to insert at least the 23.4% of `<eol>`. We assume this phenomenon is caused by the lower frequency of `<eol>` in the corpus, since a subtitle can be composed of only one line, as well as by the higher difficulty in placing the break for which the system cannot resort to speech clues (e.g. pauses).

Discussion. So far, our results indicate the higher effectiveness of *Multimodal* segmentation to automatically turn existing ST corpora into SubST-compliant training data. Also, at least for the Western European languages considered in our experiments, our approach can be successfully applied in zero-shot settings involving languages not present in the training data used to build the segmenter. While probably unrealistic (and hard to verify due to the lack of suitable benchmarks), the possibility of porting our approach to scenarios involving different alphabets is not verified in this work. This would require, at least, a vocabulary adaptation which represents a well-known problem in multilingual approaches to MT/ST (Garcia et al., 2021). Nevertheless, even in the worst case in which some degree of similarity across languages is required for zero-shot automatic segmentation, we believe that these results indicate a viable path towards overcoming the scarcity of SubST resources. In the next section, we will test this hypothesis.

4 SubST with Synthetic Data

Since our multimodal segmenter achieves the best performance overall, we use it to automatically generate the synthetic counterpart of the En→Es and En→Nl sections of MuST-Cinema. The resulting data are respectively used to train two SubST systems. The goal is to achieve comparable performance to that of similar models trained on manually segmented subtitles. For this purpose, using the same architecture, we also train two systems on the original manual segmentations of MuST-Cinema.

As shown in Table 2, the SubST system trained on our automatically segmented data (*Synthetic*) shows comparable performance with the system trained on the original segmentation (*Original*). The BLEU_{nb} between the two models is identical for Es, while for Nl the difference is not significant. On the contrary, the BLEU for the system trained on manual segmentations is higher than for the synthetic ones. These results highlight that the breaks introduced by a non-perfect automatic segmentation influence the way the subtitle breaks are placed in the translation but not necessarily the translation itself. For the length constraint, both systems obtain high CPL conformity, with the *Synthetic* model scoring 2.5% less on Es and 3.5% more on Nl. This is related to the number of `<eol>` and `<eob>` inserted by the system: the more subtitle breaks are present, the more fine-grained is the segmentation, leading to higher conformity. Indeed, CPL is higher when the Break Coverage is high.

Data	BLEU	BLEU_{nb}	CPL	EOL	EOB
Dutch					
Original	**38.6***	**25.3**	91.2%	-36.8%	+8.0%
Synthetic	36.7	24.3	**94.7%**	**-20.4%**	**+4.8%**
Spanish					
Original	**41.5***	**30.7**	**96.7%**	**-10.0%**	+10.9%
Synthetic	40.5	**30.7**	94.2%	-21.5%	**+9.9%**

Table 2: Results of the SubST systems. The * stands for statistically significant results according to bootstrap resampling test (Koehn, 2004).

Analysis. Upon examination of the segmentation patterns of the two En→Es systems,[8] we did not identify particular differences. Specifically, the inserted `<eob>` tags follow punctuation marks in 76% of the cases for both models and are followed by prepositions and conjunctions in 32% and 29% for *Original* and *Synthetic* respectively. Similar patterns between outputs were observed for `<eol>` too, which is followed by a comma in the majority of cases and by the same function words as `<eob>`. These results confirm that systems trained on automatically segmented data are able to reproduce similar segmentation patterns to those trained on original data without showing a significant degradation in the translation.

5 Conclusions

We presented an automatic segmenter able to turn existing ST corpora into SubST-compliant training data. Through comparative experiments on two language pairs in zero-shot conditions, we showed that SubST systems trained on this synthetic material are competitive with those built on human-annotated subtitling corpora. Building on these positive results, and conditioned to the availability of suitable benchmarks, verifying the portability of the approach to a larger set of languages and domains is our priority for future work.

[8] We were unable to replicate the analysis on Nl as we do not have the required linguistic competences.

References

Aitor Álvarez, Marina Balenciaga, Arantza del Pozo, Haritz Arzelus, Anna Matamala, and Carlos-D. Martínez-Hinarejos. 2016. Impact of automatic segmentation on the quality, productivity and self-reported post-editing effort of intralingual subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3049–3053, Portorož, Slovenia.

Aitor Álvarez, Carlos-D. Martínez-Hinarejos, Haritz Arzelus, Marina Balenciaga, and Arantza del Pozo. 2017. Improving the automatic segmentation of subtitles through conditional random field. *Speech Communication*, 88:83–95.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 1304–1313, New Orleans, Louisiana.

Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation. In *Proceedings of the NIPS Workshop on end-to-end learning for speech and audio processing*, Barcelona, Spain.

Mary Carroll and Jan Ivarsson. 1998. *Code of Good Subtitling Practice*.

Thierry Etchegoyhen, Lindsay Bywood, Mark Fishel, Panayota Georgakopoulou, Jie Jiang, Gerard van Loenhout, Arantza del Pozo, Mirjam Sepesy Maučec, Anja Turner, and Martin Volk. 2014. Machine Translation for Subtitling: A Large-Scale Evaluation. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, pages 46–53.

Marco Gaido, Mauro Cettolo, Matteo Negri, and Marco Turchi. 2021. CTC-based compression for direct speech translation. In *Proceedings of the 16th Conference of the European Chapter of the ACL: Main Volume*, pages 690–696, Online.

Xavier Garcia, Noah Constant, Ankur Parikh, and Orhan Firat. 2021. Towards continual learning for multilingual machine translation via vocabulary substitution. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1184–1192, Online.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 369–376, New York, NY, USA.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. Toward multilingual neural machine translation with universal encoder and decoder. *Institute for Anthropomatics and Robotics*, 2(10.12):16.

Hirofumi Inaguma, Kevin Duh, Tatsuya Kawahara, and Shinji Watanabe. 2019. Multilingual end-to-end speech translation. In *Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 570–577. IEEE.

Alina Karakanta, M. Negri, and M. Turchi. 2019. Are subtitling corpora really subtitle-like? In *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-It)*, Bari, Italy.

Alina Karakanta, Matteo Negri, and Marco Turchi. 2020a. Is 42 the answer to everything in subtitling-oriented speech translation? In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 209–219, Online.

Alina Karakanta, Matteo Negri, and Marco Turchi. 2020b. MuST-cinema: a speech-to-subtitles corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3727–3734, Marseille, France.

Alina Karakanta, Matteo Negri, and Marco Turchi. 2020c. Point Break: Surfing Heterogeneous Data for Subtitle Segmentation. In *Proceedings of the Seventh Italian Conference on Computational Linguistics (CLiC-It)*, Bologna, Italy.

Yunsu Kim, Duc Thanh Tran, and Hermann Ney. 2019. When and why is document-level context useful in neural machine translation? In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 24–34, Hong Kong, China.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain.

Maarit Koponen, Umut Sulubacak, Kaisa Vitikainen, and Jörg Tiedemann. 2020. MT for subtitling: User evaluation of post-editing productivity. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT 2020)*, pages 115–124.

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 (Demonstrations)*, pages 66–71, Brussels, Belgium.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Evgeny Matusov, Patrick Wilken, and Yota Georgakopoulou. 2019. Customizing Neural Machine Translation for Subtitling. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 82–93, Florence, Italy.

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 the 2019 Conference of the North American Chapter of the ACL (Demonstrations)*, pages 48–53, Minneapolis, Minnesota.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium.

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, Brussels, Belgium.

Umut Sulubacak, Ozan Caglayan, Stig-Arne Grönroos, Aku Rouhe, Desmond Elliott, Lucia Specia, and Jörg Tiedemann. 2020. Multimodal machine translation through visuals and speech. *Machine Translation*, 34(2):97–147.

Anke Tardel. 2020. Effort in semi-automatized subtitling processes: Speech recognition and experience during transcription. *Journal of Audiovisual Translation*, 3(2):79–102.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of Advances in Neural Information Processing Systems 30 (NIPS)*, pages 5998–6008, Long Beach, California.

Martin Volk, Rico Sennrich, Christian Hardmeier, and Frida Tidström. 2010. Machine Translation of TV Subtitles for Large Scale Production. In *Proceedings of the Second Joint EM+/CNGL Workshop "Bringing MT to the User: Research on Integrating MT in the Translation Industry (JEC'10)*, pages 53–62, Denver, CO.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020. Fairseq S2T: Fast speech-to-text modeling with fairseq. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the ACL and the 10th IJCNLP (Demonstrations)*, pages 33–39, Suzhou, China.

Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-Sequence Models Can Directly Translate Foreign Speech. In *Proceedings of Interspeech 2017*, pages 2625–2629, Stockholm, Sweden.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542, Brussels, Belgium.

A Experimental Settings

A.1 Data

For the initial experiments aimed to train textual and multimodal segmenters and select the best one (step 1 of the process described in Section 2), we use three sections of MuST-Cinema,[9] namely En→{De/Fr/It}. Each section contains paired audio utterances, transcripts, and translations, where both sides of the text are built from subtitles created by humans and therefore the subtitle breaks are based on human segmentation decisions. For French (275K sentences), German (229K sentences) and Italian (253K sentences), we used the corresponding sections of MuST-Cinema. For English, we concatenate the segmented transcripts of the previous three training sections (757K sentences).

A.2 Systems

We use the Adam optimizer and inverse square-root learning rate (lr) scheduler for all the trainings.

The **textual segmenter** is a Transformer-based (Vaswani et al., 2017) architecture consisting of 3 encoder layers and 3 decoder layers. We set the hyper-parameters as in the fairseq (Ott et al., 2019) multilingual translation task, both for the mono- and multilingual textual segmenters. For the multimodal model, a mini-batch for each language direction is built (here: 4) and the model weights are updated after each mini-batch.

The **multimodal segmenter** (Figure 1) is derived from the textual segmenter encoder-decoder structure with an additional speech encoder made of 12 Transformer encoder layers as in the original speech-to-text task[10] but with the addition of a CTC (Graves et al., 2006) module to avoid the speech encoder pre-training (Gaido et al., 2021). The training of multilingual models is realized by pre-pending the language token (`en`, `de`, `fr`, `it`) to the target sentence, as prescribed by Inaguma

[9] `https://ict.fbk.eu/must-cinema/` License: CC BY-NC-ND 4.0

[10] `https://github.com/pytorch/fairseq/tree/main/examples/speech_to_text`

Segmenter	Training	English		French		German		Italian	
		BLEU	CPL	BLEU	CPL	BLEU	CPL	BLEU	CPL
Count Chars		63.73	100%	62.04	100%	61.17	100%	60.68	100%
Textual	mono	82.6	**96.6%**	81.4	**96.7%**	79.5	90.9%	79.2	90.0%
	multi	81.8	88.5%	81.8	94.3%	80.0	90.9%	78.5	**91.2%**
Multimodal	mono	86.6	94.8%	85.5	93.9%	84.4	**91.4%**	82.1	89.9%
	multi	**87.8**	95.0%	**87.1**	94.1%	**85.4**	89.9%	**84.2**	90.0%

Table 3: Segmentation results on *seen* languages.

et al. (2019), a mechanism that was already present in the Fairseq Speech-to-text library (Wang et al., 2020). The encoder and the decoder embeddings are shared. We select the hyper-parameters of the original implementation,[11] except for the learning rate that is set to $1 \cdot 10^{-3}$, which is higher since we skipped the pre-processing phase. The vocabulary is generated using SentencePiece (Kudo and Richardson, 2018), setting the size to 10k unigrams both for the mono- and multilingual segmenters.

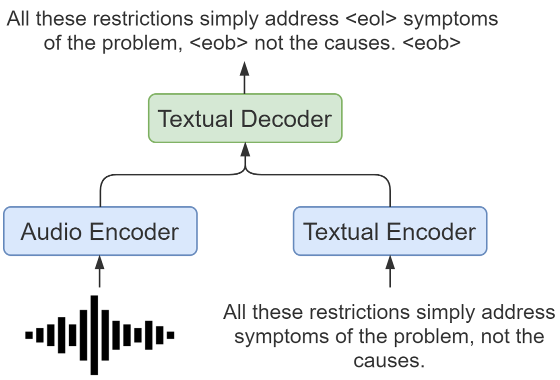

Figure 1: Multimodal segmenter architecture.

For the **supervised baseline** (*Supervised*) using OpenSubtitles data, we follow the data selection process for the highest-performing segmenter in (Karakanta et al., 2020c) (*OpenSubs- 42*). We first filter sentences with subtitles of maximum 42 characters. Since line breaks are not present in OpenSubtitles, we substitute `<eob>` symbols with `<eol>` with a probability of 0.25, paying attention not to insert two consecutive `<eol>`. This proportion reflects the `<eol>`/`<eob>` distribution featured by the MuST-Cinema training set. We noted that almost 90% of the sentences filtered contain only one subtitle. This is not very informative for the segmenter, since the only operation required is inserting one `<eob>` at the end of the sentence. For this reason, we further select only sentences with at least two subtitles (or two subtitle lines). This results in 2,956,207 sentences for Es and 683,382 sentences for Nl. We then add the same number of sentences containing only one subtitle. After this process, we obtain 5,912,414 sentences for Es and 1,366,764 sentences for Nl. The supervised baseline is trained with the same settings as the textual monolingual segmenter.

We also compare the segmenter models with a **rule-based baseline** (*Count Chars*) of inserting a break before reaching the 42-character limit, as per TED guidelines. If the 42-character limits is reached in the middle of a word, the break is inserted before this word. This method will always obtain a 100% conformity to the length constraint. As with the data filtering process, `<eol>` is inserted with probability of 0.25.

For the **SubST models** discussed in Section 4, we use the speech-to-text task *small* architecture of fairseq with the additional CTC module.

We use 4 GPUs K80 for training all the architectures: it takes around 1 day for the textual-only and around 1 week for the multimodal segmenters. However, all results are obtained by averaging 7 checkpoints (best, three preceding and three succeeding checkpoints).

B Segmentation on *seen* languages

We train the mono/multi-lingual versions of our *Textual/Multimodal* segmenters for the four language (De, En, Fr, It), measuring their performance in terms of BLEU and CPL. The results are shown in Table 3. The values of BLEU without breaks (BLEU_{nb}) are not reported since they always approach 100 i.e., the system learnt to perfectly copy the input text, as desired.

Looking at the BLEU values, both the Textual and the Multimodal segmenter perform higher than the rule-based baseline, despite a small drop in CPL. The Multimodal segmenter always outper-

[11] https://github.com/pytorch/fairseq/blob/main/examples/speech_to_text/docs/mustc_example.md

forms the Textual one by at least 5.0 BLEU points, and inserts break symbols more accurately. Moreover, it benefits from multilingual training on all languages. In contrast, overall subtitle conformity is higher for the Textual segmenter in 3 out of 4 languages, where its CPL scores are 1.2-2.6 percentage points above those obtained by the Multimodal one. Moreover, except for one case (German), higher CPL values are obtained under the monolingual training regime.