Point Break: Surfing Heterogeneous Data for Subtitle Segmentation

Alina Karakanta[1,2], **Matteo Negri**[1], **Marco Turchi**[1]
[1] Fondazione Bruno Kessler, Via Sommarive 18, Povo, Trento - Italy
[2] University of Trento, Italy
{akarakanta,negri,turchi}@fbk.eu

Abstract

Subtitles, in order to achieve their purpose of transmitting information, need to be easily readable. The segmentation of subtitles into phrases or linguistic units is key to their readability and comprehension. However, automatically segmenting a sentence into subtitles is a challenging task and data containing reliable human segmentation decisions are often scarce. In this paper, we leverage data with noisy segmentation from large subtitle corpora and combine them with smaller amounts of high-quality data in order to train models which perform automatic segmentation of a sentence into subtitles. We show that even a minimum amount of reliable data can lead to readable subtitles and that quality is more important than quantity for the task of subtitle segmentation.[1]

1 Introduction

In a world dominated by screens, subtitles are a vital means for facilitating access to information for diverse audiences. Subtitles are classified as interlingual (subtitles in a different language as the original video) and intralingual (of the same language as the original video) (Bartoll, 2004). Viewers normally resort to interlingual subtitles because they do not speak the language of the original video, while intralingual subtitles (also called captions) are used by people who cannot rely solely on the original audio for comprehension. Such viewers are, for example, the deaf and hard of hearing and language learners. Apart from creating a bridge towards information, entertainment and education, subtitles are a means to improving the reading skills of children and immigrants (Gottlieb, 2004). Having such a large pool of users and covering a wide variety of functions, subtitling is probably the most dominant form of Audiovisual Translation.

Subtitles, however, in order to fulfil their purposes as described above, need to be presented on the screen in a way that facilitates readability and comprehension. Bartoll and Tejerina (2010) claim that subtitles which cannot be read or can be read only with difficulty '*are almost as bad as no subtitles at all*'. Creating readable subtitles comes with several challenges. The difficulty imposed by the transition to a different semiotic means, which takes place when transcribing or translating the original audio into text, is further exacerbated by the limitations of the medium (time and space on screen). Subtitles should not exceed a maximum length, usually ranging between 35-46 characters, depending on screen size and audience age or preferences. They should also be presented at a comfortable reading speed for the viewer. Moreover, chucking or segmentation, i.e. the way a subtitle is split across the screen, has a great impact on comprehension. Studies have shown that a proper segmentation can balance gazing behaviour and subtitle reading (Perego, 2008; Rajendran et al., 2013). Each subtitle should – if possible – have a logical completion. This is equivalent to a segmentation by phrase, sentence or unit of information. Where and if to insert a subtitle break depends on several factors such as speech rhythm, pauses but also semantic and syntactic properties. This all makes segmenting a full sentence into subtitles a complex and challenging problem.

Developing automatic solutions for subtitle segmentation has long been impeded by the lack of representative data. Line breaks are the new lines inside a subtitle block, which are used to split a long subtitle into two shorter lines. This type of breaks is not present in the subtitle files used

to create large subtitling corpora such as OpenSubtitles (Lison and Tiedemann, 2016) and corpora based on TED Talks (Cettolo et al., 2012; Di Gangi et al., 2019), possibly because of encoding issues and the pre-processing of the subtitles into parallel sentences (Karakanta et al., 2019). Recently, MuST-Cinema (Karakanta et al., 2020b), a corpus based on TED Talks, was released, which added the missing line breaks from the subtitle files (.srt[2]) using an automatic annotation procedure. This makes MuST-Cinema a high-quality resource for the task of subtitle segmentation. However, the size of MuST-Cinema (about 270k sentences) might not be sufficient for developing automatic solutions based on data-hungry neural-network approaches, and its language coverage is so far limited to 7 languages. On the other hand, the OpenSubtitles corpus, despite being rather noisy, constitutes a large resource of subtitling data.

In this work, we leverage available subtitling resources in different resource conditions to train models which automatically segment sentences into readable subtitles. The goal is to exploit the advantages of the available resources, i.e. size for OpenSubtitles and quality for MuST-Cinema, for maximising segmentation performance, but also taking into account training efficiency and cost. We experiment with a sequence-to-sequence model, which we train and fine-tune on different amounts of data. More specifically, we hypothesise the condition where data containing high-quality segmentation decisions is scarce or non-existent and we resort to existing resources (OpenSubtitles). We show that high-quality data, representative of the task, even in small amounts, are a key to finding the break points for readable subtitles.

2 Related work

Automatically segmenting text into subtitles has long been addressed as a post-processing step in a translation/transcription pipeline. In industry, language-specific rules and simple algorithms are employed for this purpose. Most academic approaches on subtitle segmentation make use of a classifier which predicts subtitle breaks. One of these approaches used Support Vector Machine and Logistic Regression classifiers on correctly/incorrectly segmented subtitles to determine subtitle breaks (Álvarez et al., 2014). Extending this work, Álvarez et al. (2017) trained a Conditional Random Field (CRF) classifier for the same task, but in this case making a distinction between line breaks (next subtitle line) and subtitle breaks (next subtitle block). A more recent, neural-based approach (Song et al., 2019) employed a Long-Short Term Memory Network (LSTM) to predict the position of the period in order to improve the readability of automatically generated Youtube captions, but without focusing specifically on the segmentation of subtitles. Focusing on the length constraint, Liu et al. (2020) proposed adapting an Automatic Speech Recognition (ASR) system to incorporate transcription and text compression, with a view to generating more readable subtitles.

A recent line of works has paved the way for Neural Machine Translation systems which generate translations segmented into subtitles, here in a bilingual scenario. Matusov et al. (2019) customised an NMT system to subtitles and introduced a segmentation module based on human segmentation decisions trained on OpenSubtitles and penalties well established in the subtitling industry. Karakanta et al. (2020a) were the first to propose an end-to-end solution for Speech Translation into subtitles. Their findings indicated the importance of prosody, and more specifically pauses, to achieving subtitle segmentation in line with the speech rhythm. They further confirmed the different roles of line breaks (new line inside a subtitle block) and subtitle block breaks (the next subtitle appears on a new screen); while block breaks depend on speech rhythm, line breaks follow syntactic patterns. All this shows that subtitle segmentation is a complex and dynamic process and depends on several and varied factors.

3 Methodology

This section describes the data processing, model and evaluation used for the experiments. All experiments are run for English, as the language with the largest amount of available resources, but the approach is easily extended to all languages. Note that here we are focusing on a monolingual scenario, where subtitle segmentation is seen as a sequence-to-sequence task of passing from English sentences without break symbols to English sentences containing break symbols.

[2] http://zuggy.wz.cz/

3.1 Data

As training data we use MuST-Cinema and OpenSubtitles. MuST-Cinema contains special symbols to indicate the breaks: <eob> for subtitle breaks and <eol> for line breaks inside a subtitle block. We train models using all data (*MC-all*) and only 100k sentences (*MC-100*).[3]

The monolingual files for OpenSubtitles come in XML format, where each subtitle block forming a sentence is wrapped in XML tags. We are therefore able to insert the <eob> symbols for determining the end of a subtitle block. However, we mentioned that line breaks are not present in OpenSubtitles. We hence proceed to creating artificial annotations for <eol>. We filter all sentences for which all subtitles have a maximum length of 42 characters (*OpenSubs-42*). Then, for each <eob>, we substitute it with <eol> with a probability of 0.25, making sure to avoid having two consecutive <eol>, as this would lead to a subtitle of three lines, which occupies too much space on the screen. Since this length constraint results in filtering out a lot of data, we also relax the length constraint by allowing sentences with subtitles with up to 48 characters (*OpenSubs-48*). The motivation for this relaxation is that, if a sequence-to-sequence model is not able to learn the constraint of length from the data but instead learns segmentation decisions based on patterns of neighbouring words, having more data will increase the amount and variety of segmentation decisions observed by the model. This may result in more plausible segmentation, possibly though to the expense of length conformity. Dataset sizes are reported in Table 1.

We are interested in the real application scenario where high-quality data containing human segmentation decisions are not available or scarce. According to our hypothesis, a relatively limited size of high-quality data can be compensated by OpenSubtitles. Therefore, we fine-tune each of the OpenSubtitle models on 10k and 100k sentences from MuST-Cinema, which contain high-quality break annotations.

OpenSubtitles and TED Talks have been shown to have large differences and to constitute a subclassification of the subtitling genre (Müller and Volk, 2013). For this reason, we experiment with 2 test sets for cross-domain evaluation. The first set is the English test set released with MuST-Cinema, containing 10 single-speaker TED Talks (545 sentences). The second test set (782 sentences) is much more diverse. In order to create it, we have selected a mix of public and proprietary data, more specifically, excerpts from a TV series, a documentary, two short interviews and one advertising video. The subtitling was performed by professional translators and the .srt files were processed to insert the break symbols in the positions where subtitle and line breaks occur.

[3]Training a model with 10k data did not bring good results.

Data	Sents
MuST-Cinema	275,085
OpenSubs-42	185,758
OpenSubs-48	13,713,708

Table 1: Dataset sizes in sentences.

3.2 Model

The model is a sequence-to-sequence model based on the Transformer architecture (Vaswani et al., 2017), trained using fairseq (Ott et al., 2019) with the same settings as in Karakanta et al. (2020b). It takes as input a full sentence and returns the same sentence annotated with subtitle and line breaks. We process the data into sub-word units with SentencePiece (Kudo and Richardson, 2018) with 8K vocabulary size. The special symbols are kept as a single sub-word. Models were trained until convergence, on 1 Nvidia GeForce GTX1080Ti GPU.

As baseline, we use a simple segmentation approach inserting a break symbol at the first space before every 42 characters. From the two types of symbols, <eol> is selected with a 0.25 probability, but we avoid inserting two consecutive <eol>, since this would lead to a subtitle of three lines.

3.3 Evaluation

Evaluating the subtitle segmentation is performed with the following metrics. First, we compute the precision, recall and F1-score between the output of the segmenter and the human generated subtitles in order to test the model's performance at inserting a sufficient number of breaks and at the right positions in the sentence. Additionally, we compute the BLEU score (Papineni et al., 2002) between the output of the segmenter and the human reference. Higher values for BLEU indicate a high similarity between the model's and desired output.

Model	BLEU	Prec	Rec	F1	CPL	Time
baseline	55.30	50	47	48	100	-
MC-all	**84.00**	85	85	**85**	**96**	305
MC-100	81.77	84	83	83	94	**210**
OpenSubs-42	72.24	86	66	73	74	270
MC-10	77.99	83	76	79	88	+26
MC-100	80.09	87	78	81	88	+250
OpenSubs-48	76.00	77	67	68	72	6980
MC-10	82.46	86	80	82	91	+240

Table 2: Results for the MuST-Cinema test set. Training time in minutes.

Model	BLEU	Prec	Rec	F1	CPL	Time
baseline	51.45	46	43	44	100	-
MC-all	**66.38**	72	64	**69**	**97**	305
MC-100	65.38	76	64	68	96	**210**
OpenSubs-42	61.41	84	56	65	79	270
MC-10	63.53	76	60	66	93	+26
MC-100	65.3	77	62	67	94	+250
OpenSubs-48	63.37	63	56	59	81	6980
MC-10	65.66	78	61	67	94	+240

Table 3: Results for the second test set. Training time in minutes.

Finally, we want to check the performance of the system in generating readable subtitles, therefore, we use an intrinsic, task-specific metric. We compute the number of subtitles with a length of $<= 42$ characters (Characters per Line - CPL), according to the TED subtitling guidelines. This shows the ability of the system to segment the sentences into readable subtitles, by producing subtitles that are not too long to appear on the screen. We additionally report training time, as efficiency and cost are important factors for scaling such methods to tens of languages.

4 Results

Tables 2 and 3 show the results for the MuST-Cinema and the second test set respectively. As expected, the simple baseline achieves a 100% conformity to the length constraint, it is however not accurate in inserting the breaks at the right positions, as shown by the very low BLEU (55.30 and 51.45) and F1 scores (48 and 44). The best performance for all metrics and both test sets is achieved when using all available MuST-Cinema data (*MC-all*). For the in-domain test set, BLEU and F1 are higher than for the out-of-domain test set, however the number of subtitles conforming to the length constraint is consistently high (96% and 97%). This suggests that the systems trained on high-quality segmentation are able to produce readable subtitles in terms of length in diverse testing conditions even without massive amounts of data. Even with 100k of training data (*MC-100*) the performance of the model, which is the fastest model to train, drops only slightly, with -2% for all metrics on the MuST-Cinema test set and -1% on the second test set. This shows that high efficiency can be achieved without dramatically sacrificing quality. This is particularly important for industry applications where tens of languages are involved and training data for a domain might not be vast.

The models trained only on OpenSubtitles show a great drop in performance for the MuST-Cinema test, which is to be expected because of the different nature of the data. However, the drop is present also for the second test set, which shows that these models are not robust to different domains. Surprisingly, the larger model (*OpenSubs-48*) does not perform much better than the model with less data (*OpenSubs-42*) even though it is trained on almost 10 times as much data. This could be an indication of a trade-off between data quality and data size. *OpenSubs-48* with more noisy data has similar recall to *OpenSubs-42*, but it is much less accurate in the position of the breaks, as shown by the drop in precision (86 vs. 77 and 84 vs. 63). We conjecture that the procedure of artificially inserting <eol> symbols by changing the existing <eob> does not reflect the distribution of the type of breaks in real data. Interestingly, the *OpenSubs-42* model, despite containing only subtitles of a maximum length of 42, is not able to generate subtitles which respect the length constraint (74% and 79%). It is therefore possible that the segmenter does not learn to take into consideration the constraint of length, but the segmentation decisions are based on lexical patterns in the data, as also suggested by Karakanta et al. (2020a).

Fine-tuning, even on a minimum amount of real data, as shown when fine-tuning on 10k of MuST-Cinema, can significantly boost the performance compared to the OpenSubtitles models and is a viable and fast solution towards readable subtitles. This corroborates the claim in favour of creating datasets which are representative of the task at hand. Surprisingly though, fine-tuning the *OpenSubs-42* model on MC-100 does not improve over training the model from scratch on MC-100 for neither test set. For the case when only a small amount of MuST-Cinema data is available (MC-

10), having a larger base model on which to fine-tune (*OpenSubs-48*) is beneficial, since there is an improvement for all metrics and in both testing conditions compared to all other models trained on OpenSubtitles or fine-tuned on them. Therefore, we conclude that, in the presence of little data containing human segmentation decisions, a model trained or more data, even though possibly noisier, is a more robust base on which to fine-tune using the high-quality data. One considerable drawback is that the improvement comes at a training time of x25 over the other base model (*OpenSubs-42*), which raises significant considerations for cost and efficiency. Such a model however, once trained, could be re-used for fine-tuning on several domains and for different client specifications.

5 Analysis and Discussion

We further perform a manual inspection to identify issues related to the models. We hypothesise that low precision is connected to over-splitting or splitting in wrong positions, while low recall suggests under-splitting (not inserting a sufficient number of breaks). Indeed, we observe that the OpenSubtitle models tend to over-segment short sentences, but under-segment longer sentences:

Reference:
Let's turn our attention to the hows. <eob>
(37 characters)
OpenSubs-42:
Let's turn our attention <eol>
to the hows. <eob> (25 + 12 characters)

Reference:
My family's traditions <eol>
and expectations for a woman <eob>
wouldn't allow me to own a mobile <eol>
phone until I was married. <eob>
(22 + 28 + 39 + 20 characters)
OpenSubs-42:
My family's traditions and expectations <eol>
for a woman wouldn't allow me to own a mobile phone until I was married. <eob>
(39+72 characters)

In the following example, fine-tuning on MC increases length conformity, splitting the first subtitle in two, while MC-100k succeeds in segmenting all subtitles exceeding 42 characters, matching the reference segmentation.

Reference:
Meditation is a technique <eol>
of finding well-being <eob>
in the present moment <eol>
before anything happens. <eob>
OpenSubs-42:
Meditation is a technique of finding well-being <eob>
in the present moment before anything happens. <eob>
(47+46 characters)
OpenSubs-42 + MC 10K:
Meditation is a technique <eol>
of finding well-being <eob>
in the present moment before anything happens. <eob>
(25+21+46 characters)
MC-100K: Meditation is a technique <eol>
of finding well-being <eob>
in the present moment <eol>
before anything happens. <eob>

The examples above confirm our results which showed that the models do not explicitly learn the constraint of length, but rather patterns of segmentation. From a syntactic point of view, the break symbols are inserted after a noun (e.g. attention, expectations) and before a preposition/conjunction (to, for, in, before), regardless of the model. The break symbols, even though do not overlap with the human segmentation decisions, are inserted at plausible positions. This leads in subtitles that present logical completion, *i.e.* each subtitle is formed by a phrase or syntactic unit, even though they do not respect the constraint of length. The conformity to the length constraint seems to be forced only with the high-quality MuST-Cinema data. It is possible that the artificial break symbols in OpenSubtitles clash with the real break symbols in MuST-Cinema, which creates confusion for the model. Replacing some <eob> with <eol> symbols in OpenSubtitles to simulate data where human-annotated line breaks exist means that the models trained on OpenSubtitles observe a line break at positions where normally a subtitle break is present. Given the different functions of the two types of breaks, this is a possible

explanation why fine-tuning OpenSubtitles-42 on MC-100 performs worse than training on MC-100 from scratch and provides us with insights on future design of artificial segmentation decisions to augment subtitling data.

6 Conclusion

We have presented methods to combine heterogeneous subtitling data in order to improve automatic segmentation of subtitles. We leverage large data containing noisy segmentation decisions from OpenSubtitles and combine them with smaller amounts of high-quality data from MuST-Cinema to generate readable subtitles from full sentences. We found that even limited data with reliable segmentation can improve performance. We conclude that quality matters more than size for determining the break points between subtitles.

Acknowledgments

This work is part of the "End-to-end Spoken Language Translation in Rich Data Conditions" project,[4] which is financially supported by an Amazon AWS ML Grant.

References

Aitor Álvarez, Haritz Arzelus, and Thierry Etchegoyhen. 2014. Towards customized automatic segmentation of subtitles. In *Advances in Speech and Language Technologies for Iberian Languages*, pages 229–238, Cham. Springer International Publishing.

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. In *Speech Communication*, volume 88, pages 83–95. Elsevier BV.

E. Bartoll and A. Martínez Tejerina. 2010. The positioning of subtitles for the deaf and hard of hearing. *Listening to Subtitles. Subtitles for the Deaf and Hard of Hearing*, pages 69–86.

Eduard Bartoll. 2004. Parameters for the classification of subtitles. *Topics in Audiovisual Translation*, 9:53–60.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web Inventory of Transcribed and Translated Talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May.

[4] https://ict.fbk.eu/units-hlt-mt-e2eslt/

Mattia Antonino Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a multilingual speech translation corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Minneapolis, MN, USA, June.

Henrik Gottlieb. 2004. Language-political implications of subtitling. *Topics in Audiovisual Translation*, 9:83–100.

Alina Karakanta, Matteo Negri, and Marco Turchi. 2019. Are Subtitling Corpora really Subtitle-like? In *Sixth Italian Conference on Computational Linguistics, CLiC-It*.

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, July. Association for Computational Linguistics.

Alina Karakanta, Matteo Negri, and Marco Turchi. 2020b. Must-cinema: a speech-to-subtitles corpus. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*, Marseille, France, May 13-15.

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: System Demonstrations*, pages 66–71, Brussels, Belgium, November. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from Movie and TV subtitles. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC*.

Danni Liu, Jan Niehues, and Gerasimos Spanakis. 2020. Adapting end-to-end speech recognition for readable subtitles. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 247–256, Online, July. Association for Computational Linguistics.

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, August. Association for Computational Linguistics.

Mathias Müller and Martin Volk. 2013. Statistical machine translation of subtitles: From opensubtitles to ted. In Iryna Gurevych, Chris Biemann, and Torsten Zesch, editors, *Language Processing and Knowledge in the Web*, pages 132–138, Berlin, Heidelberg. Springer Berlin Heidelberg.

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*.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Elisa Perego. 2008. Subtitles and line-breaks: Towards improved readability. *Between Text and Image: Updating research in screen translation*, 78(1):211–223.

Dhevi J. Rajendran, Andrew T. Duchowski, Pilar Orero, Juan Martínez, and Pablo Romero-Fresco. 2013. Effects of text chunking on subtitling: A quantitative and qualitative examination. *Perspectives*, 21(1):5–21.

Hye-Jeong Song, Hong-Ki Kim, Jong-Dae Kim, Chan-Young Park, and Yu-Seop Kim. 2019. Inter-sentence segmentation of YouTube subtitles using long-short term memory (LSTM). 9:1504.

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 6000–6010.