Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 3069–3078
Marseille, 20-25 June 2022

Evaluating Subtitle Segmentation for End-to-end Generation Systems

Alina Karakanta[1,2], **François Buet**[3], **Mauro Cettolo**[1], **François Yvon**[3]
[1] Fondazione Bruno Kessler, Via Sommarive 18, Povo, Trento - Italy
[2] University of Trento, Italy
[3] Université Paris-Saclay, CNRS, LISN, France
{akarakanta,cettolo}@fbk.eu, {buet,yvon}@limsi.fr

Abstract

Subtitles appear on screen as short pieces of text, segmented based on formal constraints (length) and syntactic/semantic criteria. Subtitle segmentation can be evaluated with sequence segmentation metrics against a human reference. However, standard segmentation metrics cannot be applied when systems generate outputs different than the reference, e.g. with end-to-end subtitling systems. In this paper, we study ways to conduct reference-based evaluations of segmentation accuracy irrespective of the textual content. We first conduct a systematic analysis of existing metrics for evaluating subtitle segmentation. We then introduce $Sigma$, a new Subtitle Segmentation Score derived from an approximate upper-bound of BLEU on segmentation boundaries, which allows us to disentangle the effect of good segmentation from text quality. To compare $Sigma$ with existing metrics, we further propose a boundary projection method from imperfect hypotheses to the true reference. Results show that all metrics are able to reward high quality output but for similar outputs system ranking depends on each metric's sensitivity to error type. Our thorough analyses suggest $Sigma$ is a promising segmentation candidate but its reliability over other segmentation metrics remains to be validated through correlations with human judgements.

Keywords: Subtitling, Segmentation, Evaluation, Metric

1. Introduction

Accessibility of audiovisual content has become a legal obligation for major TV channels in many countries (EU, 2010) and is also a strong suggestion for uploaded web content (EU, 2016). Subtitles are a means for providing accessibility services, either with intralingual closed captioning for the deaf and hard-of-hearing or with interlingual subtitling in various languages for persons without knowledge of the source language speech. Subtitles are also useful for online talks and educational content and facilitate the comprehension of speech by language learners. Automatising the generation of subtitles has been a long-standing issue (Piperidis et al., 2004; Melero et al., 2006; Volk et al., 2010), and is nowadays more and more often performed with neural models trained end-to-end (Lakew et al., 2019; Liu et al., 2020).

Automatic generation of subtitles is a difficult task, since subtitles should not only reflect the spoken content, but should also satisfy multiple formal requirements related to their position on screen, the text length, size and colour, their display duration and synchronization with speech, etc. Additionally, a good segmentation of the transcribed or translated text into subtitles must satisfy syntactic and semantic constraints, since a segmentation which respects linguistic units has been shown to facilitate comprehension and to lead to more readable subtitles (Perego, 2008; Rajendran et al., 2013). Subtitling must thus go well beyond the automatic transcription and translation of the soundtrack, and complete automation would typically require additional processing modules such as text simplification and segmentation into readable chunks, speaker diarization and sound event detection.

In this work, we study ways to evaluate the quality of the output segmentations delivered by end-to-end subtitling systems. Contrary to pipeline systems, which typically contain an independent segmentation module that can be evaluated as a standalone component by simply measuring its ability to reproduce a reference segmentation of a reference text (*perfect text*), end-to-end systems directly output a segmented text, which may not correspond to the reference subtitle (*imperfect text*). Separating text from segmentation errors thus becomes an issue. Recent proposals to address this problem notably include metrics such as BLEU_{br} and TER_{br} (Karakanta et al., 2020b), *inter alia*, which include segmentation tags in the computation of the overall output quality. However, their ability to single out segmentation errors remains unclear. In this work, we perform a systematic assessment of subtitle segmentation metrics, with the aim to better understand their behaviour in relation to textual errors.

Our contributions can be summarised as follows:

- A comparison of existing sequence segmentation metrics to evaluate subtitle quality in the ideal situation of a *perfect* textual content, exactly matching the reference (Sec. 3.1);
- A new score $Sigma$, derived from an estimation of an upper bound of BLEU_{br}, which isolates the segmentation signal irrespective of text quality, for *imperfect* texts (Sec. 3.2);
- A boundary projection method which maps the subtitle breaks from hypothesis to the reference and allows for applying the standard segmentation metrics even for *imperfect* texts (Sec. 3.3);

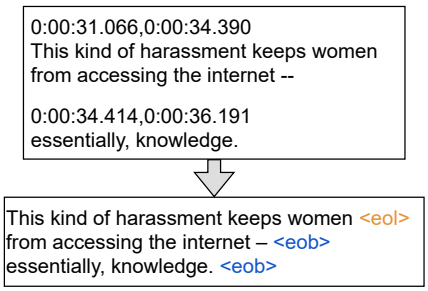


Figure 1: Example of the <eol>, <eob> segmentation notation for two subtitle blocks.

- EvalSub: A tool for computing reference-based segmentation scores for automatic subtitles.[1]

2. Generating and evaluating subtitle segmentation

2.1. Problem statement

In this work, we only focus on the evaluation of segmentation, and consider that the system's output is composed of text interspersed with segmentation symbols. We further assume that there are two types of symbols: <eol>, which indicates a change of line within the same screen, and <eob>, which indicates the end of a subtitle block and a subsequent change of screen. Figure 1 displays an example of two subtitle blocks using these notations. According to subtitling guidelines (BBC, 2021; Netflix, 2021; TED, 2021), there should be no more than two lines on the same screen, and each line should contain about 40 characters, with variations depending on the language and audience. To ease readability, line and subtitle breaks should be positioned so as to preserve the syntactic and semantic units as much as possible (Carroll and Ivarsson, 1998). In addition, the display duration of each subtitle should vary according to the number of characters on screen (reading speed), while keeping in sync with the spoken content.

Matching these constraints is a challenging task, and human subtitlers often have to compromise one constraint over the other. Moreover, the decision of where to segment depends not only on the syntax/semantics of the text (Cintas and Remael, 2007, p. 172) but also on multimodal factors such as speaker changes, speech pauses and shot changes. This means that a lot of technical, linguistic and extra-linguistic expertise goes into the segmentation process, and that there is a lot to learn from corpora of actual subtitles, be they intralingual or interlingual.

2.2. Metrics for segmentation

As for evaluating automatic segmentation, there are two possible approaches: one is to separately evaluate how well each constraint is matched, then derive an aggregate score; the other is to try to reproduce human reference segmentations. Both have their strengths and weaknesses: the former is difficult due to the need to perform a syntactic and semantic analysis of the subtitles, and to correctly weight the importance of each of the constraints listed above; the latter is tricky because standard string comparison metrics are not appropriate for subtitles. For instance, minor changes such as adding one extra <eol> can yield an invalid display with three lines. Conversely, missing one <eol> can make the current line overly long to even fit the screen. As for the position of the boundaries, it may also happen that moving a break three words ahead can better match the syntactic constraint than moving it just one word ahead.

We focus here on *reference-based metrics* and further require that a good segmentation metric should:

- account for different types of boundaries;
- accommodate scenarios where multiple human references are available;
- handle content differences with the reference;
- disentangle the effect of a poor content from that of a poor segmentation;
- realise a fair balance between all the formal and structural constraints.

We thus conduct three experiments where we 1) analyse standard segmentation metrics for evaluating subtitling segmentation and discuss how well each of them accommodates the above criteria for perfect texts (Section 3.1), 2) propose *Sigma*, a new score derived from BLEU_{br}, which allows for disentangling the effect of text quality from segmentation (Section 3.2) and 3) compare all metrics on real subtitling tasks for outputs generated by end-to-end neural machine translation and speech translation systems (Section 3.3).

3. Experimental setting

3.1. Metric sensitivity/robustness

In the first experiment, we investigate the behaviour of standard segmentation metrics and metrics previously used in the evaluation of subtitle segmentation for perfect texts in an artificial environment where we control the degree of drop in segmentation quality. The metrics are the following:

- **Precision, recall and F1** (Álvarez et al., 2016): **Precision** is defined as the proportion of boundaries in the hypothesis that agree with the reference boundaries over the total number of hypothesis boundaries, while **recall** is the number of correct boundaries divided by the reference boundaries. **F1** is the harmonic mean of precision and recall.

[1] Our code to replicate the experiments is available at https://github.com/fyvo/EvalSubtitle.

- **Window-based metrics**: $\mathbf{P}_k$ (Beeferman and Berger, 1999) assigns penalties for each moving window if ends are detected to be in different segments between reference and hypothesis, while **WindowDiff** (Pevzner and Hearst, 2002) assigns a penalty if the number of boundaries in each window is different for reference and hypothesis.

- **Edit distance-based metrics**: **Segmentation similarity** (Fournier and Inkpen, 2012) computes the proportion of boundaries that are not transformed when comparing segmentations using edit distance as a penalty function. **Boundary similarity** (Fournier, 2013) is an adaptation of segment similarity, where different weights are applied for each edit type. In $\mathbf{TER}_{\mathrm{br}}$ (Karakanta et al., 2020b) all words except boundary symbols in each hypothesis-reference pair are masked and TER (Snover et al., 2006) is computed over the masked sequences.

- $\mathbf{BLEU}_{\mathrm{br}}$ (Karakanta et al., 2020b): BLEU computed on text containing subtitle boundaries as special symbols. It has often been reported together with $\mathrm{BLEU}_{\mathrm{nb}}$ (no boundaries), computed over the hypothesis-output without the boundary symbols.

To investigate the sensitivity/robustness of the segmentation metrics in subtitling tasks, we perform changes in the reference segmentation in a controlled way. Specifically, we apply the following operations randomly on the reference segmentation: 1) shift, where a boundary is shifted 1, 2 or 3 positions to its left/right, 2) addition, where a new boundary is added to the segment, 3) deletion, where a boundary is deleted from the segment, and 4) replacement, where a boundary is substituted with the other boundary type, e.g. <eol> substituted with <eob>. For each operation type, we gradually increase the percentage of boundaries affected by the operation (20%, 40%, 60% 80% and 100%). For example, shift.1.20 corresponds to 20% of the reference boundaries shifted by one position, while delete.80 means that 80% of the reference boundaries are deleted. Additions are made with respect to the number of boundaries and not to the number of possible insertion positions (spaces); that is, add.100 doubles the number of boundaries. Finally, metrics are computed between the modified test set and the true reference.

3.2. $\mathbf{BLEU}_{\mathrm{br}}$: Content vs. segmentation

In the second experiment, we explore whether $\mathrm{BLEU}_{\mathrm{br}}$ really captures segmentation quality. BLEU without ($\mathrm{BLEU}_{\mathrm{nb}}$) and with ($\mathrm{BLEU}_{\mathrm{br}}$) boundaries have sometimes been reported together (e.g. in (Karakanta et al., 2020b) and (Buet and Yvon, 2021)), the motivation being that $\mathrm{BLEU}_{\mathrm{nb}}$ should evaluate the content, and $\mathrm{BLEU}_{\mathrm{br}}$ the segmentation. Yet, the relationship between these two scores suggests that this interpretation may be oversimplistic, motivating deeper analyses.

$\mathrm{BLEU}_{\mathrm{br}}$ is computed on longer sequences than $\mathrm{BLEU}_{\mathrm{nb}}$, which means more n-grams to match. Since predicting the right number and type of segmentation tags is generally easier than predicting the actual words, $\mathrm{BLEU}_{\mathrm{br}}$ usually has a higher unigram precision, which can, in turn, impact the higher-order precision scores.[2] This suggests that the absolute or relative difference between the two scores cannot be a proper signal of segmentation quality alone: interpreting $\mathrm{BLEU}_{\mathrm{br}} > \mathrm{BLEU}_{\mathrm{nb}}$ as a sign of good segmentation may be correct, but the intensity of this signal cannot be realistically assessed from these two measures alone. How to compare $\mathrm{BLEU}_{\mathrm{nb}}$ and $\mathrm{BLEU}_{\mathrm{br}}$? When $\mathrm{BLEU}_{\mathrm{nb}}$=100, as with perfect texts, $\mathrm{BLEU}_{\mathrm{br}}$ cannot be greater; in that case, decreases of $\mathrm{BLEU}_{\mathrm{br}}$ directly reflect segmentation errors. With imperfect texts though, the more $\mathrm{BLEU}_{\mathrm{nb}}$ goes down, the easier it is to observe $\mathrm{BLEU}_{\mathrm{br}}$ values greater than $\mathrm{BLEU}_{\mathrm{nb}}$.

Differences between $\mathrm{BLEU}_{\mathrm{br}}$ and $\mathrm{BLEU}_{\mathrm{nb}}$, as shown in Figure 2, result from matches obtained for the n-grams containing a segmentation tag (henceforth n-*tagrams*). A first possible way to disentangle the effects of the segmentation would thus be to separately compute two scores: $\mathrm{BLEU}_{\mathrm{nb}}$ and $\mathrm{BLEU}_{\mathrm{em}}$,[3] where $\mathrm{BLEU}_{\mathrm{em}}$ only measures precision scores with respect to n-tagrams. Unigram precision only counts segmentation tags, bigram precision counts 2-tagrams such as "w <tag>" or "<tag> w" etc. However, $\mathrm{BLEU}_{\mathrm{em}}$ remains highly correlated with $\mathrm{BLEU}_{\mathrm{nb}}$. This is because n-tagram matches for $\mathrm{BLEU}_{\mathrm{em}}$ directly depend on the precision of $(n-1)$-grams for $\mathrm{BLEU}_{\mathrm{nb}}$. For instance, "$\mathrm{w}_1$ w_2 <tag>" can only be correct if "w_1 w_2" is also a match, implying that the n-gram scores define an upper bound on the possible $n+1$-tagram matches.

We therefore discard $\mathrm{BLEU}_{\mathrm{em}}$ and consider instead an upper bound of $\mathrm{BLEU}_{\mathrm{br}}$, denoted $\mathrm{BLEU}_{\mathrm{br}}{}^{+}$ and computed as follows. We denote p_1, p_2, p_3, p_4 respectively as the 1g, 2g, 3g and 4g modified precisions computed by $\mathrm{BLEU}_{\mathrm{nb}}$, α the ratio of the number of boundaries to the number of word tokens, and $p'_1, \ldots, p'_4$ the corresponding precisions for $\mathrm{BLEU}_{\mathrm{br}}$. Under the assumption that boundaries are mostly correct, the expected number of correct unigrams in a text of l words augmented with boundaries is just $p_1 \times l + \alpha \times l$, yielding $p'_1 = \frac{p_1+\alpha}{1+\alpha}$. For higher order n-grams, the exact computation is more involved, but a simple upper bound is the following:

$$p'_n \leq \frac{(1-(n-1)\alpha) \times p_n + n\alpha \times p_{n-1}}{1+\alpha}.$$

This holds because we assume that: (a) each boundary

[2] Almost all our simulations have a higher unigram precision for $\mathrm{BLEU}_{\mathrm{br}}$ than for $\mathrm{BLEU}_{\mathrm{nb}}$.

[3] This approach is investigated by Élise Michon, personal communication with the authors.

REF	the car has just left Paris <eol> for its destination London <eob> where it will arrive next Sunday <eol> if all goes well . <eob>
HYP	the car has just left Paris <eol> for his destination : London <eob> where he arrives <eol> next Sunday if <eol> all goes well . <eob>

The bigram "left Paris" and trigram "all goes well" are counted both by BLEU_{nb} and BLEU_{br}; the bigram "London <eob>" and the trigram "Paris <eol> for" are counted by BLEU_{br} but not by BLEU_{nb}; conversely the bigram "if all" and trigram "arrive next Sunday" are counted by BLEU_{nb}, but not by BLEU_{br}.

Figure 2: Comparing BLEU_{nb} and BLEU_{br}

is part of n n-tagrams, within which it is surrounded by regular tokens;[4] and (b) the text is sufficiently long so that we can make the approximation $\frac{l}{l+1} \approx 1$. We readily derive an upper bound $\text{BLEU}_{\text{br}}{}^{+}$ of BLEU_{br} that we can compute from BLEU_{nb}. This value can be used as a proxy to the best achievable BLEU_{br} score for a given BLEU_{nb}. We thus denote our new score $Sigma$ (S) as:

$$S = \frac{\text{BLEU}_{\text{br}}}{\text{BLEU}_{\text{br}}{}^{+}} \quad (1)$$

Values close to 100 should signal a good segmentation, while values close to 0 a bad segmentation, *irrespective of the value of* BLEU_{nb}.

We empirically investigate the above assumptions in two steps. We first explore the relation between BLEU_{br} and BLEU_{nb} for imperfect system outputs when the segmentation remains constant. To simulate these outputs, we insert noise in the reference text, without affecting the type or position of boundaries. The noising process consists of applying a mix of edit operations (insertions, deletions, substitutions, in equal shares) corresponding to a certain percentage of the number of tokens (from 0 to 90, with a step of 10). We then move to the case when textual errors in imperfect texts are combined with segmentation errors. We apply the segmentation changes (mix of operations on the boundaries, following the same procedure as for the words) to the noisy references generated in the first step and compare the behaviour of our new score $Sigma$ compared to BLEU_{br} for different values of BLEU_{nb}.

3.3. Boundary projection

In the third experiment, we move from the scenario of controlled text and segmentation changes to investigating the usefulness of $Sigma$ for evaluating the output of real end-to-end subtitling systems. To this aim, we compare $Sigma$ to BLEU_{br} and TER_{br}, as well as to the scores obtained by standard segmentation metrics. To overcome the fact that standard metrics cannot be computed on imperfect texts, we apply a boundary projection method based on reference-hypothesis alignment, illustrated in Figure 3. Given a reference-hypothesis segment pair $\text{Ref}(1,..,i)$ and $\text{Hyp}(1,..,j)$, where i and j are respectively the number of subtitles in the reference and hypothesis segment, we split the reference and hypothesis at the subtitle boundaries, such that each subtitle (or subtitle line) is one segment. Then, the reference subtitles are aligned to the hypothesis subtitles using the MWER algorithm (Matusov et al., 2005). After this process we obtain a new reference $\text{Ref}_{proj}(1,..,j)$, containing the text of the true reference but with the boundaries projected from the hypothesis. Since Hyp and Ref_{proj} have the same number of subtitles, the boundaries of the hypothesis are simply copied in the Ref_{proj}. Projecting the boundaries from the hypothesis to the reference allows us to compute standard segmentation metrics between the projected reference Ref_{proj} and the true reference Ref, as in Experiment 3.1.

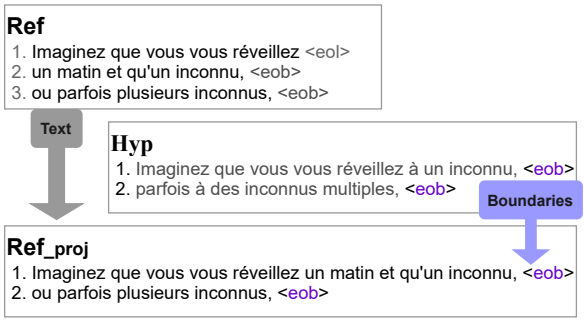


Figure 3: Projection of boundaries from hypothesis to reference based on subtitle alignment.

For comparison with previous work, we apply the boundary projection on the outputs of the 4 systems of Karakanta et al. (2020b) for En→Fr. The systems are a neural MT system (NMT), a cascade speech translation system (Cas), and two end-to-end ST systems: e2e_{base} for an ST system trained only on MuST-Cinema and e2e_{pt} for a ST system pretrained on large amounts of ST data and fine-tuned on MuST-Cinema. We then compute the segmentation metrics and discuss how the ranking of system outputs based on $Sigma$ scores differs wrt. 1) standard segmentation metrics applied on the projected reference and 2) the BLEU_{br} and TER_{br} computed between the output (without projection) and the true reference.

[4] In theory there might be several breaks in one n-tagram, but this would imply very short lines and rarely appears in our references.

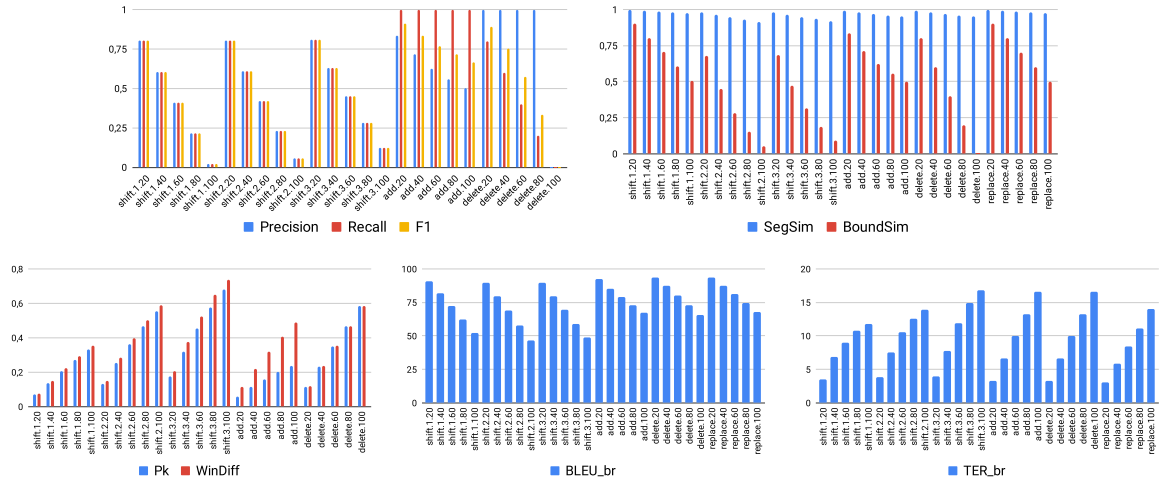


Figure 4: Behaviour of segmentation metrics when gradually transforming the reference segmentation with shift.1, shift.2, shift.3, add, delete, and replace operations applied on the boundaries. Metrics that do not distinguish different boundary types were not computed in the "replace" scenario.

3.4. Data and implementation

The subtitling data we use in the experiments come from the MuST-Cinema corpus (Karakanta et al., 2020c). The test set is compiled from the subtitle files of 9 TED talks, amounting to 545 sentences with subtitle boundaries marked as special symbols. For experiments 3.1 and 3.2 we use the English side of the English-French pair, while for the boundary projection method we use the French side.

Our code for computing the segmentation metrics is implemented in Python, based on existing libraries. The window-based metrics (P_k, WindowDiff), as well as Segment and Boundary Similarity, are computed using the SegEval package[5] (Fournier, 2013). BLEU and TER are computed with SacreBLEU[6] (Post, 2018).

4. Results

4.1. How robust/sensitive are metrics to segmentation changes?

We evaluate here the impact of several types and levels of segmentation noise on the segmentation metrics of Section 3.1. Apart from the desiderata for a good segmentation metric in Section 2.1, the types of noise we apply can have an impact on user experience: Shifts correspond to having the 'correct' number of subtitles but segmented in a non-optimal way for comprehension. Moreover, near misses are less likely to match reference boundaries, since subtitles rarely contain only 1 token (in our reference, subtitles contain on average 5 tokens[7]). Therefore, a shift of 3 positions is more likely to move the boundary in a position where another boundary is placed. For this reason, an optimal metric for subtitling segmentation should not be sensitive to shifting distance by penalising near misses less. As for additions, deletions and replacements, all of them may lead to critical errors; a deletion will lead to overly long subtitles, an addition to shorter subtitles and, as with replacements, to multiple lines in the case of consecutive <eol>. Since there are no studies clearly showing the effect of each operation on user experience, we prefer a metric which would equally penalise over- and under-generation of boundaries as well as the generation of the wrong type of boundary. Results for the degradation scenarios are shown in Figure 4. Correlation between the metrics is analysed through Pearson correlation coefficients (PCC), shown in Figure 5.

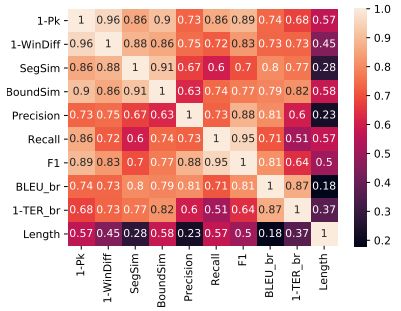


Figure 5: Pearson correlation matrix for the segmentation metrics. Coefficients were computed over the values measured in Experiment 1. "Length" is the percentage of lines conforming to the length constraint of max. 42 characters per line.

For precision-recall, shifting the boundaries causes the highest drops, despite the fact that the same number of boundaries is preserved. Interestingly, shifts by 1 position are worse than 2 positions, and in turn worse than

[5] https://pypi.org/project/segeval/

[6] BLEU|#:1|c:mixed|e:no|tok:13a|s:exp|v:2.0.0
TER|#:1|c:lc|t:tercom|nr:no|pn:yes|as:no|v:2.0.0

[7] We acknowledge that this number may vary for languages with different scripts and subtitling conventions.

3 positions. The difference is more visible for percentages above 60%. F1 deteriorates more for deletions than for additions because of a stronger drop in recall.

By design, the error measured by P_k is always lower than that measured by WindowDiff (an error for P_k is an error for WindowDiff, but the reverse is not true). As noted and criticised by Pevzner and Hearst (2002), P_k penalises false negatives (FNs) heavier than false positives (FPs) (in our experiment FNs correspond to deletions, and FPs to additions). Thus P_k appears to be more recall-oriented than the other metrics, which is confirmed by the higher PCC with the recall metric. However, for P_k and WindowDiff penalties increase regularly with shift size and reversely to precision and recall (1<2<3). Again, these metrics are more sensitive to deletions than to additions.

SegSim computes cosmetically high values (as mentioned by Fournier (2013)), which can be inconvenient for interpretation since it lacks sufficient resolution. The new normalisation introduced for BoundSim notably solves this issue. As with the window-based metrics, SegSim and BoundSim are more sensitive to deletions and additions and give less penalty to near misses (here only for shifts of 1). This can be explained by the fact that shifting a boundary by 1 position is accounted as one transposition, while longer shifts cost one addition and one deletion. However, TER_{br} is robust to the type of error, as it shows a balance between deletions and additions, as well as shifts of 3 positions. All edit-based metrics are less sensitive to replacements.

$BLEU_{br}$ globally remains within the 45–100 range, since it is the only metric considering textual content. It is hardly sensitive to shift size; shifting 1, 2 or 3 positions yields almost the same scores, but shifts are more penalised than the other types of noise. Despite being a precision-based metric, $BLEU_{br}$ here is actually robust to error type, since it equally penalises deletions, additions, as well as replacements. It is worth considering that one wrongly omitted boundary will affect the n-gram precision (for $n > 1$), although not as much as one wrongly added boundary. Therefore, this balance between deletion and addition penalisation could be attributed to the effect of brevity penalty, which decreases the score of segments with missing boundaries.

To conclude, drawing back to our criteria for a good segmentation metric: the ability to account for different types of boundaries is present in SegSim, BoundSim, $BLEU_{br}$, and TER_{br}. $BLEU_{br}$ and TER_{br} can take advantage of multiple human references. As for the balance between formal and structural constraints, even though all metrics highly correlate with each other, their correlation with the length conformity (see "Length" in Figure 5) is low, with only recall, P_k and BoundSim correlating above 0.5. For this, precision-recall metrics could give some insights into the type of error (over- or under-segmentation), but should be computed separately for each type of break and combined. This suggests that there is still a long way to go for a segmentation metric that incorporates formal and structural subtitling constraints.

Apart from the factors mentioned above, sensitivity to near misses is not a desired property for a subtitle segmentation metric. Unlike window- and edit-based metrics, $BLEU_{br}$ is in line with our expectations regarding the shifting distance of boundaries. It also achieves balance between the additions, deletions and replacements. Therefore, $BLEU_{br}$ seems to have properties corresponding to our criteria for a good segmentation metric. Still, our expectations on the relationship of metrics with user experience has to be investigated with user studies and the final decision on a good segmentation metric can be taken only after validating its correlation with human judgements.

4.2. What does $BLEU_{br}$ really measure?

In spite of its flaws, $BLEU_{nb}$ remains the go-to metric for MT research and studies on automatic subtitling. As for its relation to segmentation quality, in Section 3.1, we showed that for perfect output text ($BLEU_{nb}$=100) $BLEU_{br}$ indeed ranks segmentation from good to bad (fourth graph of Figure 4), with the added benefit of yielding a global aggregate value taking into account all types of errors (additions, deletions, replacements). The question we now turn to is the usefulness of $BLEU_{br}$ *for imperfect texts*. We would like to independently evaluate quality in content prediction (with $BLEU_{nb}$) and quality in segmentation prediction. Can $BLEU_{br}$, as used in previous work, or $Sigma$ (equation (1)), play that role?

First, for 'perfect' segmentation but imperfect text, Figure 6 shows that the relationship between $BLEU_{br}$ and $BLEU_{nb}$ is linear. This indeed confirms our hypothesis that the two metrics are so correlated that their difference cannot be a strong signal of segmentation quality. It also shows that $BLEU_{br}$ cannot exceed an upper bound which is strongly related to $BLEU_{nb}$, since in this setting the segmentation has not been affected by noise. Given this dependency between $BLEU_{br}$ and $BLEU_{nb}$, reporting both scores is not informative and the segmentation signal should be sought in a different relationship.

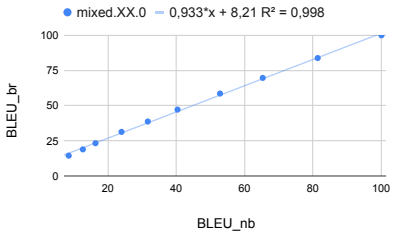
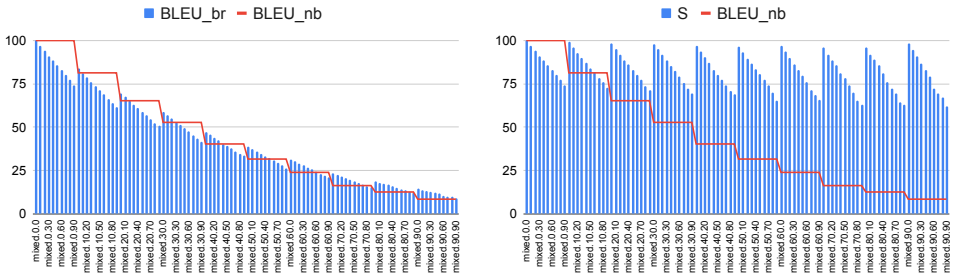


Figure 6: Linearity of $BLEU_{br}$ wrt $BLEU_{nb}$, for instances where only text was submitted to noise (cf. Section 3.2). Linear regression gives a coefficient of determination of 0.998, and a standard error of 1.2.



(a) (b)

Figure 7: Values of BLEU_{br} (a) and $Sigma$ (b) after applying segmentation noise at different values of BLEU_{nb} (10 levels of segmentation noise for each BLEU_{nb} value, 10 BLEU_{nb} values from 100 down to 8.5). In (a) we see BLEU_{br} decreasing with decreasing BLEU_{nb}, whereas (b) $Sigma$ scores remain stable.

When moving to the scenario of imperfect text and imperfect segmentation, Figure 7a confirms our hypothesis that the more the quality of BLEU_{nb} drops, the more often BLEU_{br} exceeds BLEU_{nb}. On the contrary, $Sigma$ remains in a similar range irrespective of the value of BLEU_{nb}. We illustrate this in Figure 7b where we plot $Sigma$ for various values of BLEU_{nb}. $Sigma$ responds linearly to the amount of segmentation noise, but we observe a minor drift of the range of values when BLEU_{nb} decreases (from [74.3–100] for mixed.0 to [63.2-95.8] for mixed.90). However, in a realistic scenario, BLEU_{nb} would typically be constrained to an interval between 25 and 55 (corresponding to the 4 central series on Figure 7b). Moreover, when comparing the segmentation ability of two systems close in BLEU_{nb} value, the impact of that drift would be all the more limited. This shows that $Sigma$ can be a good approximate value for capturing segmentation quality, irrespective of the quality of the generated content.

4.3. Boundary projection

We here move to the evaluation of the segmentation of actual outputs from end-to-end systems. Since standard metrics cannot be applied to imperfect texts, we use the boundary projection method as a proxy to compare the system ranking based on all metrics to that of $Sigma$. Scores are reported in Table 1. $Sigma$ ranks the output of `NMT` as best with a score of 89.2, followed by `Cas` with 83.1, while the two direct systems score very closely with 81.8 for $\texttt{e2e}_{base}$ and 81.5 for $\texttt{e2e}_{pt}$. In relation to the standard metrics on the projected reference (columns 2-10) and the metrics considering text quality (BLEU_{br} and TER_{br} to the left of $Sigma$), all metrics clearly agree with $Sigma$ on their rating of the output of the `NMT` system as having the best segmentation among the examined outputs. In TED talks, subtitlers create the target subtitles (our reference) using the source subtitles as template, therefore it is expected that a system receiving the source boundaries as input and able to correctly copy the boundaries will achieve high similarity with the reference. However, when comparing the scores for the three Speech Translation systems (`Cas`, $\texttt{e2e}_{base}$ and $\texttt{e2e}_{pt}$), the agreement among metrics is lost. The cascade output seems to have better segmentation with 6 wins (WindowDiff, SegSim, precision, BLEU_{br} and the two versions of TER_{br}) and 3 ties (P_k, F1 and BLEU_{br} on system output) over the $\texttt{e2e}_{pt}$, which is ranked best according to BoundSim and recall. The close scores in many metrics, including $Sigma$, show no clear winner between the two direct systems. This is expected since $\texttt{e2e}_{pt}$ was pretrained on non-segmented text, which improves translation quality, but did not receive any additional segmentation data compared to $\texttt{e2e}_{base}$. The length of the predicted output seems to have an effect, since metrics with a high correlation with length conformity rank $\texttt{e2e}_{pt}$ higher than `Cas` (length conformity is 95% for $\texttt{e2e}_{pt}$ and 91% for `Cas`). All in all, the two ST systems seem to be making different types of errors (`Cas` has higher precision, $\texttt{e2e}_{pt}$ higher recall), but for most metrics the scores are so close that it may be hard to tell which output is the best.

The results show that, even though the metrics are capable of properly rewarding high quality output, distinguishing between outputs of similar segmentation quality under real evaluation settings is a difficult task and requires metrics with sufficient resolution. Despite this, $Sigma$ shows a relatively high agreement with the majority of metrics in ranking the cascade output as best among the ST outputs. The boundary projection method is used here as a proxy for disentangling the effect of text quality from segmentation, but the scores computed through this method are impacted by the performance of the alignment algorithm, especially for low quality outputs. On the other hand, metrics computed directly on imperfect texts (BLEU_{br} and TER_{br}) are strongly influenced by translation quality, as shown by the different ranking of BLEU_{br} computed on projected reference (column 9) vs. system output (column 11). $Sigma$ is not constrained by either of these limitations and provides a clear, interpretable and easy-to-compute solution for evaluating the segmentation even between imperfect texts of similar quality.

System	P_k	Windiff	SegSim	BndSim	Prec	Rec	F1	$BLEU_{br}$	TER_{br}	$BLEU_{br}$	TER_{br}	S
NMT	.192	.208	.979	.637	.711	.735	.723	83.18	6.87	32.16	19.38	89.2
Cas	<u>.252</u>	<u>.270</u>	<u>.970</u>	.519	<u>.639</u>	.667	<u>.653</u>	<u>76.14</u>	<u>8.91</u>	26.34	<u>23.23</u>	<u>83.1</u>
$e2e_{base}$	.257	.277	.969	.515	.601	.667	.632	75.00	9.29	22.53	24.48	81.8
$e2e_{pt}$	<u>.252</u>	.276	.969	<u>.525</u>	.610	<u>.702</u>	<u>.653</u>	74.89	9.24	<u>26.36</u>	23.52	81.5

Table 1: Segmentation scores of $Sigma$ and the examined metrics for the output of four systems after projecting their boundaries to the reference (Section 4.3). The two columns to the left of $Sigma$ are $BLEU_{br}$ and TER_{br} scores between the output and the reference without projection. Best score among the ST outputs is underlined.

5. Related work

Automatic subtitle segmentation has been previously evaluated in the case of interlingual and intralingual subtitles, by comparing the automatically generated output against a reference (Álvarez et al., 2014). For interlingual subtitles, Álvarez et al. (2016) proposed a segmentation algorithm based on Logistic regression and Support Vector Machine classifiers. The evaluation was performed with precision-recall-F1 measures, based on the ability of the algorithm to insert a segmentation boundary, without however distinguishing between line and subtitle breaks. Later, Álvarez et al. (2017) compared rule-based and machine-learning segmentation methods with metrics which considered either subtitle breaks only or both line and subtitle breaks. These metrics count the segmentation errors (F1, NIST), the number of incorrectly segmented portions (DSER) or the edit distance between sequences of reference positions and hypothesis positions (SegER). Karakanta et al. (2020a) trained a sequence-to-sequence model on different combinations of real and synthetically segmented data, which transforms an unsegmented sentence into a sentence with line and subtitle breaks. Except for F1, performance is evaluated with BLEU as a similarity measure, and characters per line (CPL), as the percentage of segmented subtitles conforming to the length constraint.

Segmentation has also been evaluated in the context of Machine Translation for subtitling. Karakanta et al. (2020b) report $BLEU_{br}$, where the BLEU metric of Papineni et al. (2002) is computed on text containing subtitle boundaries annotated as special symbols. Each boundary symbol counts as one token in the BLEU computation. Similarly, Matusov et al. (2019) report scores for their MT metrics in the *S-mode*, where line breaks in a subtitle are marked with a separator symbol. However, in their setting there is a one-to-one correspondence between source-target subtitles. Karakanta et al. (2020b) also introduce TER_{br}, a variant of TER (Snover et al., 2006) which is computed on text where all words except for the subtitle boundaries are masked. The authors claim that this metric determines the effort required by a human subtitler to manually correct the segmentation, ignoring word errors. Last, Cherry et al. (2021) propose two metrics again related to BLEU: 1) *Timed-BLEU*, where target to reference alignments necessary for the evaluation are created by linear temporal alignment, over which BLEU is calculated as usual, and 2) *T-BLEU Headroom*, calculated as the difference between an upper bound of T-BLEU and the actual T-BLEU. Both metrics only apply when the output contains timestamps, which are not always available in subtitle generation with end-to-end systems.

6. Conclusion

We have analysed metrics and methods to evaluate the segmentation of text into subtitles, given a human reference. Our analysis using artificial noise in segmentation has shown that for perfect texts, $BLEU_{br}$ satisfies our criteria for a good subtitle segmentation metric. However, when moving to imperfect texts, $BLEU_{br}$ correlates highly with regular BLEU, therefore the segmentation signal cannot be extracted by a simple difference between $BLEU_{br}$ and $BLEU_{nb}$. We thus introduce a new subtitle segmentation score $Sigma$, as the ratio of $BLEU_{br}$ to its approximated upper bound. In order to compare $Sigma$ with standard segmentation metrics for evaluating real system outputs, we further proposed a boundary projection method which projects the subtitle boundaries from the output to the reference. We noted that in real evaluation settings existing metrics do not always agree on their ranking of the outputs, especially for outputs of similar quality. We believe that the final response on the most accurate method to evaluate subtitle segmentation can be given only after obtaining correlations of the metrics and methods proposed in this paper with human judgements. However, the analysis presented in this work has shed light into the critical aspects of evaluating subtitle segmentation, in order to better design user studies to collect these human judgements and to refine the approximation of upper bound $BLEU_{br}$ for computing $Sigma$ in subtitling segmentation tasks.

7. Acknowledgements

This work was partially supported by the European Commission funded project "Humane AI: Toward AI Systems That Augment and Empower Humans by Understanding Us, our Society and the World Around Us" (grant #820437) and by the BPI-France investment programme "Grands défis du numérique", as part of the ROSETTA-2 project (Subtitling RObot and Adapted Translation). The support is gratefully acknowledged.

8. Bibliographical References

Álvarez, A., Arzelus, H., and Etchegoyhen, T. (2014). Towards customized automatic segmentation of subtitles. *Advances in Speech and Language Technologies for Iberian Languages*, pages 229–238.

Álvarez, A., Balenciaga, M., del Pozo, A., Arzelus, H., Matamala, A., and Martínez-Hinarejos, C.-D. (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, May. European Language Resources Association (ELRA).

Álvarez, A., Martínez-Hinarejos, C.-D., Arzelus, H., Balenciaga, M., and del Pozo, A. (2017). Improving the automatic segmentation of subtitles through conditional random field. *Speech Communication*, 88:83–95.

BBC. (2021). Subtitling guidelines, version 1.1.9. Available online: `https://bbc.github.io/subtitle-guidelines/`.

Beeferman, D. and Berger, A. (1999). Statistical models for text segmentation. *Machine learning*, 34:177–210.

Buet, F. and Yvon, F. (2021). Toward Genre Adapted Closed Captioning. In *Interspeech 2021*, pages 4403–4407, Brno (virtual), Czech Republic, August. ISCA.

Carroll, M. and Ivarsson, J. (1998). *Code of Good Subtitling Practice*. Simrishamn: TransEdit.

Cherry, C., Arivazhagan, N., Padfield, D., and Krikun, M. (2021). Subtitle translation as markup translation. In *Proceedings of INTERSPEECH 2021*.

Cintas, J. D. and Remael, A. (2007). *Audiovisual Translation: Subtitling*. Translation practices explained. Routledge.

EU, E. U. (2010). Audiovisual Media Services Directive, Directive 2010/13/EU of the European Parliament and of the Council. Available online: `http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32010L0013&from=EN`. Last accessed: 20 December 2021.

EU, E. U. (2016). EU Directive on Web Content Accessibility, Directive 2016/2102/EU of the European Parliament and of the Council on the accessibility of the websites and mobile applications of public sector bodies. Available online: `https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016L2102&from=EN`. Last accessed: 06 January 2021.

Fournier, C. and Inkpen, D. (2012). Segmentation similarity and agreement. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 152–161, Montréal, Canada, June. Association for Computational Linguistics.

Fournier, C. (2013). Evaluating text segmentation using boundary edit distance. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1702–1712, Sofia, Bulgaria, August. Association for Computational Linguistics.

Karakanta, A., Negri, M., and Turchi, M. (2020a). Point Break: Surfing Heterogeneous Data for Subtitle Segmentation. In *Seventh Italian Conference on Computational Linguistics, CLiC-It*.

Karakanta, A., Negri, M., and Turchi, M. (2020b). 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.

Karakanta, A., Negri, M., and Turchi, M. (2020c). MuST-cinema: a speech-to-subtitles corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3727–3734, Marseille, France, May. European Language Resources Association.

Lakew, S. M., Gangi, M. D., and Federico, M. (2019). Controlling the output length of neural machine translation. In *Proceedings of IWSLT'2019*.

Liu, D., Niehues, J., and Spanakis, G. (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.

Matusov, E., Leusch, G., Bender, O., and Ney, H. (2005). Evaluating Machine Translation Output with Automatic Sentence Segmentation. In *Proceedings of the Second International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA, October 24-25.

Matusov, E., Wilken, P., and Georgakopoulou, Y. (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.

Melero, M., Oliver, A., and Badia, T. (2006). Automatic multilingual subtitling in the eTITLE project. *Proceedings of Translating and the Computer*, 28:1–18.

Netflix. (2021). Timed text style guide: General requirements. Available online: `https://partnerhelp.netflixstudios.com/hc/en-us/articles/215758617-Timed-Text-Style-Guide-General-Requirements`. Last accessed: 10/06/2021.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (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, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Pevzner, L. and Hearst, M. A. (2002). A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

Piperidis, S., Demiros, I., Prokopidis, P., Vanroose, P., Hoethker, A., Daelemans, W., Sklavounou, E., Konstantinou, M., and Karavidas, Y. (2004). Multimodal, multilingual resources in the subtitling process. In *Proceedings of LREC*.

Post, M. (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, October. Association for Computational Linguistics.

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

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the seventh conference of the Association for Machine Translation in the America (AMTA)*, pages 223–231, Boston, Massachusetts, USA.

TED. (2021). Subtitling tips. Available online: `https://www.ted.com/participate/translate/subtitling-tips`.

Volk, M., Sennrich, R., Hardmeier, C., and Tidström, F. (2010). Machine translation of TV subtitles for large scale production. In *Second Joint EM+/CNGL Workshop*, pages 53–62, November.