Direct Segmentation Models for Streaming Speech Translation

Javier Iranzo-Sánchez, Adrià Giménez Pastor, Joan Albert Silvestre-Cerdà,
Pau Baquero-Arnal, Jorge Civera, Alfons Juan
Machine Learning and Language Processing (MLLP) research group
Valencian Research Institute for Artificial Intelligence (VRAIN)
Universitat Politcnica de Valncia, Spain
{jairsan,adgipas,jsilvestre,pabaar,jorcisai,ajuanci}@vrain.upv.es

Abstract

The cascade approach to Speech Translation (ST) is based on a pipeline that concatenates an Automatic Speech Recognition (ASR) system followed by a Machine Translation (MT) system. These systems are usually connected by a segmenter that splits the ASR output into, hopefully, semantically self-contained chunks to be fed into the MT system. This is specially challenging in the case of streaming ST, where latency requirements must also be taken into account. This work proposes novel segmentation models for streaming ST that incorporate not only textual, but also acoustic information to decide when the ASR output is split into a chunk. An extensive and thorough experimental setup is carried out on the Europarl-ST dataset to prove the contribution of acoustic information to the performance of the segmentation model in terms of BLEU score in a streaming ST scenario. Finally, comparative results with previous work also show the superiority of the segmentation models proposed in this work.

1 Introduction

ST is a field that is very closely aligned with ASR and MT, as it is their natural evolution to combine the advances in both areas. Thus, the goal is to obtain the translation of an utterance that has been spoken in a different language, without necessarily requiring the intermediate transcription. At the same time, it is desirable to have high quality translations without compromising the speed of the system. Although research into ST started in the nineties (Waibel et al., 1991), the field did not really take off until significant breakthroughs were achieved in ASR (Chan et al., 2016; Irie et al., 2019; Park et al., 2019; Jorge et al., 2020) and MT (Bahdanau et al., 2015; Sennrich et al., 2016b,a; Vaswani et al., 2017), mainly due to the introduction of deep neural networks (NN). Thanks to this, the field has recently attracted significant amounts of attention from both the research and industry communities, as the field is now mature enough that it has tangible and well-performing applications (Ma et al., 2019; Jia et al., 2019).

Currently, there are two main approaches to ST: cascade and end-to-end models. The goal of the end-to-end approach is to train a single system that is able to carry out the the entire translation process (Weiss et al., 2017; Berard et al., 2018; Gangi et al., 2019). This has only recently been possible thanks to advances in neural modeling. Due to a lack of ST data, different techniques such as pre-training and data augmentation (Bahar et al., 2019; Pino et al., 2019) have been used in order to alleviate this lack of data. It is important to remark that the currently proposed end-to-end models work in an offline manner and must process the entire input sequence. Therefore they cannot be used for a streaming scenario.

In the cascade approach, an ASR system transcribes the input speech signal, and this is fed to a downstream MT system that carries out the translation. The provided input to the MT step can be the 1-best hypothesis, but also n-best lists (Ng et al., 2016) or even lattices (Matusov and Ney, 2011; Sperber et al., 2019). Additional techniques can also be used to improve the performance of the pipeline by better adapting the MT system to the expected input, such as training with transcribed text (Peitz et al., 2012) or chunking (Sperber et al., 2017). The cascade approach can be used to take advantage of independent developments in ASR and MT, and it is significantly easier to train due to greater data availability. Thus, it is very relevant to study improvements for the ST cascade pipeline.

This work focuses on the effects of segmentation in streaming ST, as cascade systems still outperform end-to-end systems in standard setups (Niehues et al., 2019; Pino et al., 2019). Fol-

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 2599–2611,
November 16–20, 2020.

lowing a cascade approach, a streaming ST setup can be achieved with individual streaming ASR and MT components. Advances in neural streaming ASR (Zeyer et al., 2016; Jorge et al., 2019, 2020) allow the training of streaming models whose performance is very similar to offline ones. Recent advances in simultaneous MT show promise (Arivazhagan et al., 2019; Ma et al., 2019; Zheng et al., 2019), but current models have additional modelling and training complexity, and are not ready for translation of long streams of input text. For the scenario to be considered (translation of parliamentary speeches, with an average duration of 100s), it is required for the ST systems to have a minimum throughput, but simultaneous translation is not required, so the translation of chunks[1] is still acceptable. In this case we prioritize quality over simultaneous translation, with a streaming ASR system followed by a standard offline MT system. This way, the resulting ST cascade system can provide transcribed words in real-time, that are eventually split into chunks to be translated by the offline MT system.

Following this approach, it is necessary to incorporate a segmentation component in the middle in order to split the output of the ASR system into (hopefully semantically self-contained) chunks that can be successfully processed by the MT model, while maintaining a balance between latency and quality. In (Cho et al., 2012, 2015, 2017), the authors approach this problem by training a monolingual MT system that predicts punctuation marks, and then the ASR output is segmented into chunks based on this punctuation. Another approach is to segment the ASR output by using a language model (LM) that estimates the probability of a new chunk to start (Stolcke and Shriberg, 1996; Wang et al., 2016, 2019). Binary classifiers with Part of Speech and reordering features have also been proposed (Oda et al., 2014; Siahbani et al., 2018). It is also possible to segment the ASR output using handcrafted heuristics such as those based on a fixed number of words per chunk (Cettolo and Federico, 2006) or acoustic information (Fügen et al., 2007). These heuristic approaches present the disadvantage of being very domain and speaker specific. Alternatively, segmentation can be integrated into the decoder, so that it is carried out at the target side rather than the source side (Kolss et al., 2008).

[1] A chunk must be understood as a sequence of words.

This work introduces a statistical framework for the problem of segmentation in ST, which incorporates both textual and acoustic information. Jointly with this, we propose a set of novel models that follow this framework, and a series of extensive experiments are carried out, which show how these new models outperform previously proposed segmentation models. In addition, we study the effect of the preprocessing scheme applied to the input of the MT system, the performance degradation explained by transcription and/or segmentation errors, and the latency due to the components of the ST system.

This paper is organized as follows. The next section describes the statistical framework of the segmenter in the streaming ST scenario. Section 3 follows, detailing how our proposed models are instantiated in this framework. Then, Section 4 describes the Europarl-ST dataset that is used in the experiments and the three main components of our streaming ST system based on a cascade approach: ASR and MT systems, and the segmentation models. Next, Section 5 reports a detailed evaluation in terms of BLEU score on the Europarl-ST dataset and comparative results with previous work, and latency figures. Finally, Section 6 draws the main conclusions of this work and devises future research lines.

2 Statistical framework

We define the streaming ST segmentation as a problem in which a continuous sequence of words provided as the output of an ASR system is segmented into chunks. These chunks will then be translated by a downstream MT system. The goal of the segmentation is to maximize the resulting translation accuracy while keeping latency under the response-time requirements of our streaming scenario.

Formally, the segmentation problem is the task of dividing a sequence of input words x_1^J into non-overlapping chunks. We will represent this with a sequence of split/non-split decisions, y_1^J, with $y_j = 1$ if the associated word x_j is the word that ends a chunk; and $y_j = 0$, otherwise. Optionally, additional input features can be used. In this work, we use word-based acoustic features (a_1^J) aligned with the sequence of words output by the ASR system.

Ideally, we would choose the segmentation $\hat{y}_1^J$

such that,

$$\hat{y}_1^J = \arg\max_{y_1^J} p(y_1^J \mid x_1^J, a_1^J)$$
$$= \arg\max_{y_1^J} \prod_{j=1}^{J} p(y_j \mid y_1^{j-1}, x_1^J, a_1^J). \quad (1)$$

However, in a streaming setup, we need to bound the sequence to w words into the future (hereafter, *future window*) to meet latency requirements

$$\hat{y}_1^J = \arg\max_{y_1^J} \prod_{j=1}^{J} p(y_j \mid y_1^{j-1}, x_1^{j+w}, a_1^{j+w}). \quad (2)$$

Indeed, for computational reasons, the sequence is also bounded to h words into the past (hereafter, *history size*)

$$\hat{y}_1^J = \arg\max_{y_1^J} \prod_{j=1}^{J} p(y_j \mid y_{j-h}^{j-1}, x_{j-h}^{j+w}, a_{j-h}^{j+w}). \quad (3)$$

Previous works in the literature can be stated as a particular case of the statistical framework defined above under certain assumptions.

LM based segmentation (Stolcke and Shriberg, 1996; Wang et al., 2016, 2019). In this approach, an n-gram LM is used to compute the probability

$$P(y_j) = p(x_{j-n+1}^{j-1}, y_{j-n+1}^{j-1}, x_j, y_j, x_{j+1}^{j+n-1}) \quad (4)$$

where y_j is zero or one depending on the non-split or split decision to be taken, respectively. Split and non-split probabilities are combined into a function to decide whether a new chunk is defined after x_j

$$\hat{y}_j = \arg\max_{y_j} f(P(y_j)). \quad (5)$$

Monolingual MT segmentation (Cho et al., 2012, 2015, 2017). Following this setup, a monolingual MT model translates from the original, (unpunctuated) words x_1^J into a new sequence z_1^J that contains segmentation information (via puntuaction marks). Each z_j can be understood as a pair (x_j, y_j), so the segmentation can be defined as an MT problem

$$\hat{z}_1^J = \arg\max_{z_1^J} p(z_1^J \mid x_1^J), \quad (6)$$

that basically reverts to

$$\hat{y}_1^J = \arg\max_{y_1^J} p(y_1^J \mid x_1^J) \quad (7)$$

since x_1^J is given.

In contrast with previous approaches, which treats segmentation as a by-product of a more general task, we propose a model that directly represents the probability of the split/non-split decision.

3 Direct Segmentation Model

Now that we have introduced the theoretical framework, we are going to describe our proposed segmentation model. This approach has the advantage of allowing a future dependency and consider not only textual, but also acoustic features. This provides the model with additional evidence for taking a better split/non-split decision.

First, the *Text* model computes text state vectors s_j^{j+w} that consider each word in x_{j-h}^{j+w} using an embedding function $e()$ and one or more recurrent layers, represented by the function $f_1()$. In order to incorporate information about previous decisions y_{j-h}^{j-1}, we create a new sequence $\tilde{x}_{j-h}^{j+w}$ by inserting an end-of-chunk token into the text input sequence every time a split decision has been taken. This sequence is bounded in length by h.

$$\tilde{x}_{j-h}^{j+w} = f_c(x_{j-h}^{j+w}, y_{j-h}^{j-1}). \quad (8)$$

Then, the state vectors are defined as follows

$$s_j^{j+w} = f_1(e(\tilde{x}_{j-h}^{j+w}))). \quad (9)$$

Next, the split probability is computed by concatenating the state vectors of the current word and those in the future window, and passing them through a series of feedforward layers $f_2()$

$$p(y_j \mid y_{j-h}^{j-1}, x_{j-h}^{j+w}) = f_2([s_j^{j+w}]). \quad (10)$$

If we include acoustic information, acoustic state vectors are computed using function $f_3()$

$$c_j^{j+w} = f_3(a_{j-h}^{j+w}) \quad (11)$$

and are concatenated with the text state vectors in order to compute the split/non-split probability

$$p(y_j | y_{j-h}^{j-1}, x_{j-h}^{j+w}, a_{j-h}^{j+w}) = f_2([s_j^{j+w}; c_j^{j+w}]). \quad (12)$$

In the case of audio information, we assess two variants, depending whether the acoustic sequence is passed through a RNN (*Audio w/ RNN*) or not (*Audio w/o RNN*). These word-based acoustic feature vectors are obtained as follows. The Audio w/o RNN (also referred to as *copy*) option extracts

three acoustic features associated to each word: duration of the current word, duration of the previous silence (if any), and duration of the next silence (if any). The three features were selected due to effectiveness (indeed, they improve system performance) as well as being word-based features which therefore can be directly integrated into the proposed model. At training time, these features are obtained by carrying out a forced alignment between the audio and the reference transcription, while at testing time are directly provided by the ASR system. The Audio w/ RNN option adds an independent RNN as f_3, to process the sequence a_{j-h}^{j+w} of three-dimensional acoustic feature vectors just described, and the acoustic state vectors are concatenated at word level with the text state vectors. Whenever acoustic features are used, first the Text model is pre-trained and frozen, and then the feedforward network is updated with the acoustic data.

$$f_3(a_{j-h}^{j+w}) = \begin{cases} RNN(a_{j-h}^{j+w}) & \text{Audio w/ RNN} \\ a_j^{j+w} & \text{Audio w/o RNN} \end{cases} \quad (13)$$

Figure 1 provides an overview of the proposed model architecture behind the streaming ST segmenter. The part of the model inside the dashed-line boundary represents the Text model (Equations 8,9), while the complete model that additionally considers acoustic information is represented outside the boundary for the Audio w/ RNN and Audio w/o RNN cases (Equations 11, 13). State vectors are concatenated before the feed-forward network (FFN). Equation 10 computes the split probability for the Text-only model, and Equation 12 does the same for the Audio models.

4 Experimental setup

To study the effects of our streaming ST segmenter in terms of BLEU score (Papineni et al., 2002), state-of-the-art ASR and MT systems were trained to perform ST from German (De), Spanish (Es) and French (Fr) into English (En), and vice versa. ASR and MT systems were treated as black boxes in order to focus our efforts on evaluating the proposed streaming ST segmentation models on the recently released and publicly available Europarl-ST corpus (Iranzo-Sánchez et al., 2020). Basic statistics of the six language pairs of the Europarl-ST corpus involved in the evaluation are shown in Table 1.

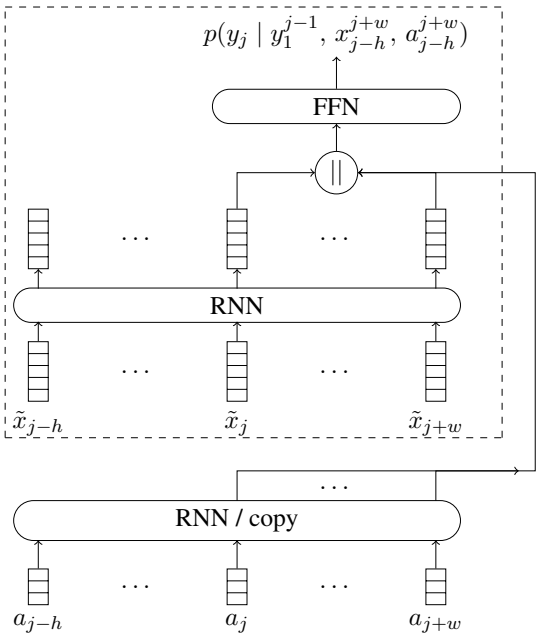

Figure 1: Overview of the model architecture for the streaming ST segmenter. The dashed-line boundary separates the Text model including word embeddings, RNN and state vectors, from the two possible Audio models, RNN and copy, outside the boundary.

4.1 ASR systems

In our cascade ST setting, input speech signal is segmented into speech/non-speech regions using a Gaussian Mixture Model - Hidden Markov Model based voice activity detection (VAD) system (Silvestre-Cerdà et al., 2012), which will be referred to as the baseline segmentation system. Detected speech chunks are delivered to our general-purpose hybrid ASR systems for German (De), English (En), Spanish (Es) and French (Fr).

On the one hand, acoustic models (AM) were generated using the TLK toolkit (del Agua et al., 2014) to train Feed-Forward deep neural Network - Hidden Markov Models (FFN-HMM). These models were used to bootstrap bidirectional long-short term memory (BLSTM) NN models (Zeyer et al., 2017), trained using Tensorflow (Abadi et al., 2015), except for the French ASR system which only features FFNs. These AMs were trained with 0.9K (De), 5.6K (En), 3.9K (Es), and 0.7K (Fr) hours of speech data from multiple sources and domains.

On the other hand, language models (LM) consist of a linear interpolation of several n-gram LMs trained with SRILM (Stolcke, 2002), combined with a recurrent NN (RNN) LM trained using the RNNLM toolkit (Mikolov, 2011) (De, Fr),

Table 1: Basic statistics of the Europarl-ST corpus for the training, development and test partitions for the six language pairs involved in the evaluation.

ST Direction	Training			Development			Test		
	Videos	Kwords		Videos	Kwords		Videos	Kwords	
		Source	Target		Source	Target		Source	Target
En-De	2937	753	730	134	29	28	126	28	27
En-Es	2926	738	800	131	29	31	127	28	31
En-Fr	2918	738	901	132	29	34	124	72	33
De-En	1082	245	289	218	50	58	226	52	59
Es-En	727	203	200	202	53	53	206	50	50
Fr-En	1053	328	395	148	39	36	166	48	45

or an LSTM LM trained with the CUED-RNNLM toolkit (Chen et al., 2016) (Es, En). The vocabulary of LMs was restricted to 200K words. As training monolingual text data, we disposed of 0.8G (De), 300G (En), 0.7G (Es) and 1.8G (Fr) tokens.

Regarding ASR performance, these systems show 19.8 (De), 17.2 (En), 10.9 (Es) and 24.3 (Fr) Word Error Rate% (WER%) figures in their corresponding test sets of the Europarl-ST corpus.

4.2 MT systems

Neural MT systems were trained for each of the translation directions to be studied using the fairseq toolkit (Ott et al., 2019). The initial models are general out-of-domain systems trained with millions (M) of sentences: 21.0M for De↔En, 21.1M for En↔Es and 38.2M for En↔Fr. These models followed the sentence-level Transformer (Vaswani et al., 2017) BASE configuration, and were fine-tuned using the Europarl-ST training data.

Two MT systems were trained for each translation direction depending on the preprocessing scheme applied to the source sentences in the training set. The first scheme uses a conventional MT preprocessing (tokenization, truecasing, etc.), while the second scheme applies a special ST preprocessing to the source side of the training set, by lowercasing, transliterating and removing punctuation marks from all sentences (Matusov et al., 2018). This latter preprocessing scheme guarantees that the same conditions for the MT input are found at training and inference time. Since conventional MT preprocessing was applied to the target side, our hope is that the model is also able to learn to recover casing and punctuation information from the source to the target side. Both preprocessing schemes were evaluated by translating ASR hypotheses provided in chunks given by the baseline VAD segmenter. Results are shown in Table 2. As the segmentation is different from that of the reference, the evaluation is carried out by re-segmenting the translations so that they match the segmentation of the reference (Matusov et al., 2005).

As shown in Table 2, BLEU score improvements of the ST scheme over the MT scheme range from 4.1 (En-De) to 7.5 (En-Es), due to the fact that the ST source processing scheme fixes the mismatch between training and inference time. At the same time, MT systems are able to recover punctuation information that was not available in the ASR output. Thus, the special ST preprocessing scheme was applied in the rest of experiments.

4.3 Segmentation models

Depending on the segmentation model, text and optionally audio belonging to Europarl-ST were used as training data. As a preprocessing step, an end-of-chunk token was inserted in the text training data after each punctuation mark, such as full point, question/exclamation marks, etc., delimiting a chunk. In addition, the ST preprocessing scheme was applied to the annotated reference transcriptions in order to obtain training data that mimics ASR output. In the case of Audio models, as mentioned before, audio and reference transcriptions were forced-aligned using the AMs described in Section 4.1 in order to compute word and silence durations as acoustic features.

Due to the class imbalance present in the segmentation problem, (95% of samples belong to the non-split class), training batches were prepared by weighted random sampling so that on average, one third of the samples belongs to the split class. Otherwise, the model degenerates to always classifying into the non-split class.

The Text model consists of 256-unit word-

Table 2: BLEU scores of the cascade ST on the Europarl-ST test sets depending on the preprocessing scheme.

Source prep. scheme	En-De	En-Es	En-Fr	Es-En	Fr-En	De-En
Conventional MT	22.4	28.0	23.4	26.5	25.4	21.3
Special ST	26.5	35.5	29.3	33.8	29.9	25.8

embedding layer, followed by a forward GRU-based RNN of 256 units. Second, for the Audio w/ RNN model, acoustic features are processed by a forward GRU-based RNN of 8 units. State vectors from Text, and optionally Audio w/ RNN, are fed into a two-layer FFN of 128 units and RELU activation. A dropout of 0.3 is applied after the RNN and FFN layers. Architecture decisions were taken on the basis of the BLEU results obtained on the dev set.

Given the sequential nature of the split/non-split decision process as a streaming ASR output is processed, greedy and beam search decoding algorithms were implemented and compared, but negligible differences were observed between them.

5 Evaluation

In order to perform an evaluation that simulates real conditions, the ASR hypothesis of an entire speech (intervention made by a MEP, with an average duration of 100 seconds) is fed to the segmentation model whose generated chunks are translated by the MT system. The chunks are translated independently from each other. The quality of the MT output, in terms of BLEU score, provides a clear indication of the performance of the streaming ST segmenter and allows us to compare different segmenters.

Figure 2 shows BLEU scores as a function of the length of the future window for the English-German (En-De) and Spanish-English (Es-En) dev sets. On the left-hand side, the three segmenters (Text, Audio w/ RNN and Audio w/o RNN) are compared averaging their BLEU scores over history sizes 5, 10 and 15 for the sake of clarity. On the right-hand side, the effect of history sizes is analysed for the Audio w/o RNN segmenter. In both cases, reference transcriptions were used as input to the segmenter.

As observed, the length of the future window is a very significant parameter to decide whether to split or not, which validates our decision to use a model that considers not only past history, but also a future window. In the case of En-De, adding a future window significantly improves the results, up to 5.8 and 3.5 BLEU points on average, in the Text and Audio models, respectively. Similarly for Es-En, but at a lower magnitude, a gain of up to 3.7 and 3.4 BLEU points on average in the Text and Audio models, respectively, is obtained at larger future windows.

When comparing the segmenters (Figure 2 on the left), the Text segmenter provides a performance that is clearly lower than the Audio-based segmenters for the English-German pair, and similar or lower for the Spanish-English pair. Audio-based segmenters offers nearly the same BLEU scores for English-German and Spanish-English. However, the Audio w/o RNN being a simpler model reaches slightly better BLEU scores using future window of length 4. This window length presents an appropriate trade-off between system latency and accuracy in our streaming scenario. Focusing on the Audio w/o RNN segmenter (Figure 2 on the right), longer history sizes such as 10 and 15 clearly provide better BLEU scores than the shorter history size ($h = 5$). A history size of 10 reaches the best BLEU scores for English-German, and similar performance is achieved between 10 and 15 in Spanish-English for future window of length 4. Based on these results, a history size of 10 and a future window of length 4 were selected for the rest of the experiments.

Table 3 presents BLEU scores of conventional cascade ST systems, in which the ASR output is segmented using the three proposed models and passed down to the MT system, from English into German, Spanish and French, and vice versa. As an upper-bound reference, results on an oracle segmenter are provided, which we have approximated by splitting the text into chunks using punctuation marks. The oracle segmenter shows which are the best BLEU scores that can be achieved with our current ASR and MT systems.

BLEU scores show how, except for Spanish-English, the models with acoustic features are able to outperform those that are only text-based. The largest improvement is in the English-German case, with a 0.8 BLEU-point improvement of Audio models over the Text model. When comparing the

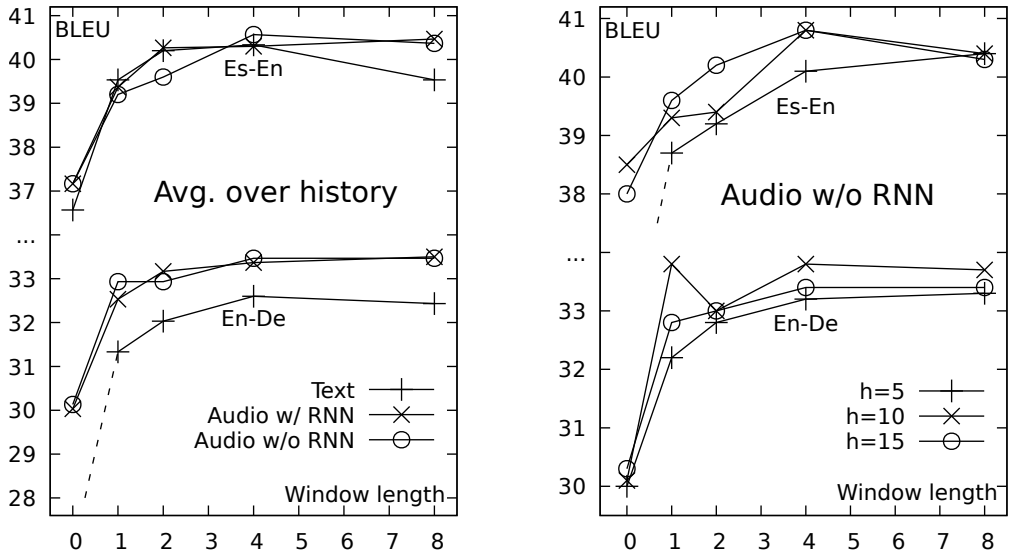

Figure 2: BLEU scores in the English-German (En-De) and Spanish-English (Es-En) dev sets as a function of future window length, averaged over history sizes for the three segmenters on the left-hand side, and on history sizes 5, 10 and 15 for the Audio w/o RNN segmenter on the right-hand side.

Table 3: BLEU scores on the test sets provided by the conventional cascade ST system with ASR output.

Segmenter	En-De	En-Es	En-Fr	Es-En	Fr-En	De-En
Baseline (VAD)	26.5	35.5	29.3	33.8	29.9	25.8
Text	27.6	37.0	29.4	34.7	31.6	28.1
Audio w/o RNN	28.4	37.2	30.0	34.4	32.1	28.3
Audio w/ RNN	28.4	37.3	30.1	33.9	32.1	28.2
Oracle	31.6	41.3	33.6	38.1	35.3	31.3

Audio models, there does not seem to be an improvement of using RNN to process the acoustic features with respect to directly feeding the acoustic features to the FFN. In the case of the Es-En, our analysis shows that, as one one set of hyperparameters was optimized and then shared among all language directions, the resulting segments produced by the Es-En Audio models are around 60% longer than segments in other pairs, which results in reduced performance of the sentence-based MT model.

Table 4 shows BLEU scores when the ASR output is replaced by the reference transcription, so that errors are only due to the segmenter and the MT systems. These results follow the trend of those in Table 3, with improvements of Audio over Text models, and no significant differences between both Audio models. Unlike the previous case, the Es-En Audio w/o RNN system does improve the results of the Text model. Interestingly enough, the oracle segmentation allows us to observe the performance degradation specifically due to the segmentation model, that is, between 2.7 and 5.3 BLEU points. Those oracle results show the best-case scenario that can be achieved with the current MT systems, using the reference transcriptions and the reference segmentation. As the addition of the RNN to process the acoustic features does not improve the performance, the simpler Audio w/o RNN will be used in the remaining experiments.

5.1 Comparison with previous work

In this section, we compare our results with previous work in the literature described in Section 2: the n-gram LM based segmenter included in the SRILM toolkit (Stolcke, 2002), and the monolingual MT segmentation (Cho et al., 2017) whose implementation is also publicly available[2].

Table 5 shows BLEU scores of a cascade ST system for the English-German Europarl-ST test set, comparing the two segmenters mentioned above, the Audio w/o RNN model proposed in this work, and the oracle segmenter that provides the refer-

[2]https://github.com/jniehues-kit/SLT.KIT

Table 4: BLEU scores on the test sets provided by a cascade ST system with reference transcriptions.

Segmenter	En-De	En-Es	En-Fr	Es-En	Fr-En	De-En
Text	33.3	43.3	35.6	37.8	38.1	30.0
Audio w/o RNN	34.2	44.2	36.2	38.2	38.8	30.3
Audio w/ RNN	34.1	44.1	36.2	37.4	38.7	30.3
Oracle	37.2	47.4	38.9	41.3	41.5	35.6

Table 5: Comparison with previous work in terms of BLEU score on the English-German test set of the Europarl-ST corpus.

Segmenter	Train data	ASR	References
LM based	EP-ST	27.0	32.9
	+ IWSLT	26.5	31.7
Mono MT	EP-ST	28.0	33.8
	+ IWSLT	28.1	34.1
This work	EP-ST	28.4	34.2
	+ IWSLT	28.5	35.0
Oracle	–	31.6	37.2

ence segmentation. Except for the oracle, these segmenters were trained using only the Europarl-ST (EP-ST) training set, or the Europarl-ST training set plus additional training data from the IWSLT 2012 evaluation campaign (Cettolo et al., 2012) , in order to study the performance of the segmenter when additional, out-of-domain text data is available. Results translating both, ASR hypotheses as well as reference transcriptions, are provided.

The results show the same trend across inputs to the MT system, ASR outputs, and reference transcriptions; but differences in BLEU over segmenters are more noticeable when segmenting the references. The LM based segmenter provides the lowest BLEU scores and is not able to take advantage of additional IWSLT training data. The monolingual MT model is at a middle ground between the LM based segmenter and our segmenter, but it is able to take advantage of the additional IWSLT training data. However, our segmenter outperforms all other segmenters in both training data settings. More precisely, when incorporating IWSLT training data, our segmenter outperforms by 0.4 BLEU (ASR output) and 0.9 BLEU (reference transcriptions) the best results of previous work obtained using the monolingual MT model, mainly thanks to the ability to use acoustic information. Additionally, our proposed model shrinks the gap with respect to the oracle segmentation to 3.1 BLEU points working with ASR output, and 2.2 BLEU points when reference transcriptions are provided.

5.2 Latency evaluation

We will now measure the latency of our cascade ST system in a streaming scenario. Following (Li et al., 2020), we define accumulative chunk-level latencies at three points in the system, as the time elapsed between the last word of a chunk being spoken, and: 1) The moment the consolidated hypothesis for that chunk is provided by the ASR system; 2) The moment the segmenter defines that chunk on the ASR consolidated hypothesis; 3) The moment the MT system translates the chunk defined by the segmenter. These three latency figures, in terms of mean and standard deviation, are shown in Table 6. It should be noticed that this ST system is working with ASR consolidated hypotheses in the sense that these hypotheses will not change as the audio stream is further processed.

The difference of 1.1 seconds between the ASR and the segmenter is mostly due to the need to wait for the words in the future window to be consolidated, as the time taken by the segmenter to decide whether to split or not is negligible ($\simeq$ 0.01s). Lastly, the MT system has a delay of 0.5 seconds. The total latency is dominated by the ASR system, since the long-range dependencies of the RNN-based LM delay the consolidation of the hypothesis, which is needed by the segmenter and the MT system in order to output the definitive translation.

In practice, however, the ST system could work with non-consolidated hypotheses, since these hypotheses very rarely change with respect to those consolidated. In this case, the latency of the ASR system is significantly reduced to 0.8 ± 0.2 seconds, while the latency experienced by the user for the whole ST system is 1.3 ± 0.4 seconds, as the segmenter does not wait for the words in the future window to be consolidated.

6 Conclusions

This work introduces a statistical framework for the problem of ASR output segmentation in stream-

Table 6: Accumulative chunk-level latencies in seconds (mean $\pm$ std. dev.) for the ASR, Segmenter and MT components of the Es-En ST cascade model.

	Latency (seconds)
ASR	4.1 ± 1.6
+ Seg.	5.2 ± 2.2
+ MT	5.7 ± 2.2

ing ST, as well as three possible models to instantiate this framework. In contrast to previous works, these models not only consider text, but also acoustic information. The experimental results reported provide two key insights. Firstly, we have confirmed how the preprocessing of the MT training data has a significant effect for ST, and how a special preprocessing that is closer to the inference conditions is able to obtain significant improvements. Secondly, we have shown the importance of including acoustic information in the segmentation process, as the inclusion of these features improves system performance. The proposed model improves the results of previous works on the Europarl-ST test set when evaluated with two training data setups.

In terms of future work, there are many ways of improving the segmenter system that has been presented here. We plan to look into additional acoustic features as well as possible ways to incorporate ASR information to the segmentation process. The segmenter model itself could also benefit from the incorporation of additional text data as well as pre-training procedures. We also devise two supplementary research lines, the integration of the segmentation into the translation process, so the system learns how to segment and translate at the same time, and moving from an offline MT system to a streaming MT system to improve response time, but without performance degradation.

Acknowledgements

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 761758 (X5Gon); the Government of Spain's research project Multisub, ref. RTI2018-094879-B-I00 (MCIU/AEI/FEDER,EU), the Generalitat Valenciana's research project Classroom Activity Recognition, ref. PROMETEO/2019/111., FPU scholarship FPU18/04135; and the Generalitat Valencianas predoctoral research scholarship ACIF/2017/055. The authors wish to thank the anonymous reviewers for their criticisms and suggestions.

References

Martín Abadi et al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org.

Miguel A. del Agua, Adrià Giménez, Nicolás Serrano, Jesús Andrés-Ferrer, Jorge Civera, Alberto Sanchis, and Alfons Juan. 2014. The Translectures-UPV Toolkit. In *Advances in Speech and Language Technologies for Iberian Languages*, pages 269–278. Springer International Publishing.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic Infinite Lookback Attention for Simultaneous Machine Translation. In *Proc. of ACL*, pages 1313–1323. Association for Computational Linguistics.

Parnia Bahar, Tobias Bieschke, and Hermann Ney. 2019. A comparative study on end-to-end speech to text translation. In *Proc. of IEEE ASRU*, pages 792–799.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proc. of ICLR*.

Alexandre Berard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-End Automatic Speech Translation of Audiobooks. In *Proc. of ICASSP*, pages 6224–6228. IEEE.

Mauro Cettolo and Marcello Federico. 2006. Text segmentation criteria for statistical machine translation. In *In Proc. of Advances in Natural Language Processing, FinTAL*, volume 4139 of *Lecture Notes in Computer Science*, pages 664–673. Springer.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT^3: Web Inventory of Transcribed and Translated Talks. In *Proc. of EAMT*, pages 261–268.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proc. of ICASSP*, pages 4960–4964. IEEE.

Xi Chen, Xin Liu, Y. Qian, Mark J. F. Gales, and Philip C. Woodland. 2016. CUED-RNNLM An open-source toolkit for efficient training and evaluation of recurrent neural network language models. In *Proc. of ICASSP*, pages 6000–6004. IEEE.

Eunah Cho, Jan Niehues, Kevin Kilgour, and Alex Waibel. 2015. Punctuation insertion for real-time spoken language translation. In *Proc. of IWSLT*. ISCA.

Eunah Cho, Jan Niehues, and Alex Waibel. 2012. Segmentation and punctuation prediction in speech language translation using a monolingual translation system. In *Proc. of IWSLT*. ISCA.

Eunah Cho, Jan Niehues, and Alex Waibel. 2017. NMT-Based Segmentation and Punctuation Insertion for Real-Time Spoken Language Translation. In *Proc. of Interspeech*, pages 2645–2649. ISCA.

Christian Fügen, Alex Waibel, and Muntsin Kolss. 2007. Simultaneous translation of lectures and speeches. *Machine Translation*, 21(4):209–252.

Mattia Antonino Di Gangi, Matteo Negri, Roldano Cattoni, Roberto Dessi, and Marco Turchi. 2019. Enhancing Transformer for End-to-end Speech-to-Text Translation. In *Proc. of MT Summit XVII Volume 1: Research Track*, pages 21–31. EAMT.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020. Europarl-ST: A Multilingual Corpus For Speech Translation Of Parliamentary Debates. In *Proc. of ICASSP*, pages 8229–8233. IEEE.

Kazuki Irie, Albert Zeyer, Ralf Schlter, and Hermann Ney. 2019. Language Modeling with Deep Transformers. In *Proc. Interspeech 2019*, pages 3905–3909. ISCA.

Ye Jia, Ron J. Weiss, Fadi Biadsy, Wolfgang Macherey, Melvin Johnson, Zhifeng Chen, and Yonghui Wu. 2019. Direct speech-to-speech translation with a sequence-to-sequence model. In *Proc. of Interspeech*, pages 1123–1127. ISCA.

Javier Jorge, Adrià Giménez, Javier Iranzo-Sánchez, Jorge Civera, Albert Sanchis, and Alfons Juan. 2019. Real-time One-pass Decoder for Speech Recognition Using LSTM Language Models. In *Proc. of Interspeech*, pages 3820–3824.

Javier Jorge, Adrià Giménez, Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Jorge Civera, Albert Sanchis, and Alfons Juan. 2020. LSTM-Based One-Pass Decoder for Low-Latency Streaming. In *Proc. of ICASSP*. IEEE.

Muntsin Kolss, Stephan Vogel, and Alex Waibel. 2008. Stream decoding for simultaneous spoken language translation. In *Proc. of INTERSPEECH*, pages 2735–2738. ISCA.

B. Li, S. Chang, T. N. Sainath, R. Pang, Y. He, T. Strohman, and Y. Wu. 2020. Towards Fast and Accurate Streaming End-To-End ASR. In *Proc. of ICASSP*. IEEE.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proc. of ACL*, pages 3025–3036. ACL.

Evgeny Matusov, Gregor Leusch, Oliver Bender, and Hermann Ney. 2005. Evaluating machine translation output with automatic sentence segmentation. In *Proc. of IWSLT*. ISCA.

Evgeny Matusov and Hermann Ney. 2011. Lattice-based ASR-MT interface for speech translation. *IEEE Trans. Audio, Speech & Language Processing*, 19(4):721–732.

Evgeny Matusov, Patrick Wilken, Parnia Bahar, Julian Schamper, Pavel Golik, Albert Zeyer, Joan Albert Silvestre-Cerdà, Adrià Martínez-Villaronga, Hendrik Pesch, and Jan-Thorsten Peter. 2018. Neural speech translation at AppTek. In *Proc. of IWSLT*, pages 104–111. ISCA.

T. Mikolov. 2011. The RNNLM Toolkit. `http://www.fit.vutbr.cz/~imikolov/rnnlm/`.

Raymond W. M. Ng, Kashif Shah, Lucia Specia, and Thomas Hain. 2016. Groupwise learning for ASR k-best list reranking in spoken language translation. In *Proc. of ICASSP*, pages 6120–6124. IEEE.

J. Niehues, R. Cattoni, S. Stker, M. Negri, M. Turchi, T. Ha, E. Salesky, R. Sanabria, L. Barrault, L. Specia, and M Federico. 2019. The IWSLT 2019 Evaluation Campaign. In *Proc. of IWSLT*. ISCA.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2014. Optimizing segmentation strategies for simultaneous speech translation. In *Proc. of ACL*, pages 551–556. ACL.

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 *Proc. of NAACL-HLT: Demonstrations*. ACL.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proc. of ACL*, pages 311–318. ACL.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Proc. Interspeech 2019*, pages 2613–2617.

Stephan Peitz, Simon Wiesler, Markus Nußbaum-Thom, and Hermann Ney. 2012. Spoken language translation using automatically transcribed text in training. In *Proc. of IWSLT*, pages 276–283. ISCA.

Juan Pino, Liezl Puzon, Jiatao Gu, Xutai Ma, Arya D McCarthy, and Deepak Gopinath. 2019. Harnessing Indirect Training Data for End-to-End Automatic Speech Translation: Tricks of the Trade. In *Proc. of IWSLT*. ISCA.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Proc. of ACL*, pages 86–96. ACL.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proc. of ACL*, pages 1715–1725. ACL.

Maryam Siahbani, Hassan Shavarani, Ashkan Alinejad, and Anoop Sarkar. 2018. Simultaneous translation using optimized segmentation. In *Proc. of AMTA*, pages 154–167. Association for Machine Translation in the Americas.

Joan Albert Silvestre-Cerdà, Adrià Giménez, Jesús Andrés-Ferrer, Jorge Civera, and Alfons Juan. 2012. Albayzin Evaluation: The PRHLT-UPV Audio Segmentation System. In *Proc. of IberSPEECH 2012*, pages 596–600.

Matthias Sperber, Graham Neubig, Ngoc-Quan Pham, and Alex Waibel. 2019. Self-attentional models for lattice inputs. In *Proc. of ACL*, pages 1185–1197. ACL.

Matthias Sperber, Jan Niehues, and Alex Waibel. 2017. Toward robust neural machine translation for noisy input sequences. In *Proc. of IWSLT*. ISCA.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proc. of Interspeech*, pages 901–904. ISCA.

Andreas Stolcke and Elizabeth Shriberg. 1996. Automatic linguistic segmentation of conversational speech. In *Proc. of ICSLP*, volume 2, pages 1005–1008. ISCA.

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 *Proc. of NIPS*, pages 5998–6008.

Alex Waibel, Ajay N Jain, Arthur E McNair, Hiroaki Saito, Alexander G Hauptmann, and Joe Tebelskis. 1991. JANUS: a speech-to-speech translation system using connectionist and symbolic processing strategies. In *Proc. of ICASSP*, pages 793–796. IEEE.

Xiaolin Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2016. An Efficient and Effective Online Sentence Segmenter for Simultaneous Interpretation. In *Proc. of WAT*, pages 139–148. The COLING 2016 Organizing Committee.

Xiaolin Wang, Masao Utiyama, and Eiichiro Sumita. 2019. Online Sentence Segmentation for Simultaneous Interpretation using Multi-Shifted Recurrent Neural Network. In *Proc. of MT Summit XVII Volume 1: Research Track*, pages 1–11. EAMT.

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

Albert Zeyer, Patrick Doetsch, Paul Voigtlaender, Ralf Schlüter, and Hermann Ney. 2017. A comprehensive study of deep bidirectional LSTM RNNS for acoustic modeling in speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, pages 2462–2466. IEEE.

Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2016. Towards online-recognition with deep bidirectional LSTM acoustic models. In *Proc. of Interspeech 2016*, pages 3424–3428. ISCA.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019. Simpler and faster learning of adaptive policies for simultaneous translation. In *Proc. of EMNLP-IJCNLP*, pages 1349–1354. ACL.

A Reproducibility

The source code of the Direct Segmentation Model, as well as the ASR hypothesis and acoustic features used in the experiments are attached as supplementary materials. Combined with the instructions provided for training the MT systems, this allows for faithful reproduction of our experiments.

B ASR Systems

The acoustic models were trained using the datasets listed on Table 7, and the architecture of the models is summarized in Table 8.

The language models were trained using the datasets listed on Table 9. The number of English words includes 294G words from Google Books counts. As for the models themselves, they are an interpolation between 4-gram LM and a RNNLM. For German and French, the RNN is trained with the RNNLM toolkit and has a hidden layer of 400 units. For Spanish and English, the RNN is a LSTM trained with the CUED toolkit, with an embbeding layer of 256 units and a hidden layer of 2048 units. The vocabulary was limited to the most common 200K words.

Table 7: Statistics of the speech resources used for acoustic model training .

English		Spanish		German		French	
Corpus	Hours	Corpus	Hours	Corpus	Hours	Corpus	Hours
Crawled Data	3313	Crawled Data	3466	Crawled Data	716	Crawled Data	592
LibriSpeech	960	PM	261	GSC-TUDa	158	TEDx	39
TED-LIUM v3	454	EPPS	157	Audiobooksfr	28		
CommonVoice	243	Voxforge	21	Voxforge	21		
SWC	154						
VL.NET	110						
Voxforge	109						
AMI	96						
EPPS	79						
ELFA	48						
VCTK	44						

Table 8: Details of the acoustic models architecture.

	English	Spanish	German	French
MFCC	80	85	48	48
Input size	80	85	48x11	48x11
Standard Model (1-pass)	8x1024(BLSTM)	8x1024(BLSTM)	6x2048(DNN)	6x2048(DNN)
Output states (1-pass)	16132	10041	18867	6282
fCMLLR model (2-pass)	–	–	5x1024(BLSTM)	6x2048(DNN)
Output states (2-pass)	–	–	18867	6651

Table 9: Statistics of text resources used for language modelling.

English		Spanish		German		French	
Corpus	MWords	Corpus	MWords	Corpus	MWords	Corpus	MWords
News-Discuss	3650	OpenSubtitles	1146	Wikipedia	642	Giga	665
Wikipedia	2266	Ufal	910	Europarl	46	Wikipedia	375
News Crawl	1120	Wikipedia	586	Comm. Crawl	45	UN	358
LibriSpeech	804	United Nations	343	News-Crawl	30	OpenSubs	263
GIGA	617	News Crawl	298	Reuters	38	DGT	79
United Nations	334	Crawled data	116	Tatoeba	3	Europarl	55
HAL	92	Comm. Crawl	41			COSMAT	29
Europarl	54					TT2	13
DGT-TM	45					News comm.	5
News comm.	6					TED	4
WIT-3	3					AMARA fr	1
COSMAT	1					EUTV	1
EuroParl TV	1						

Table 10: Satistics of the text resources used for training MT systems.

Corpus	Samples(M)		
	De-En	Fr-En	Es-En
DGT	5.1	–	–
EUbookshop	9.3	–	5.2
TildeMODEL	4.2	–	–
Wikipedia	2.4	–	1.8
UN	–	11.0	–
GIGA	–	22.5	–
newscommentary	–	1.0	–
commoncrawl	–	3.2	1.8
EU-TT2	–	–	1.0

Table 11: Segmentation model hyperparameter exploration. Selected values are shown in bold.

Hyperparameter	Values
Embedding size	128,**256**,512,1024
RNN size	128,**256**,512,1024
FF layers	1,**2**,3
FF size	128,**256**,512
Batch size	128,**256**,512
Learning rate	0.001,**0.0001**
Optimizer	**Adam**
Dropout	**0.3**,0.5
History size	0,1,2,5,**10**,15,20
Future window	0,1,2,**4**,8

C MT Systems

The models were trained using the datasets listed on Table 10.

The following fairseq command was used to train the systems:

```
fairseq-train $CORPUS_FOLDER \
 -s $SOURCE_LANG_SUFFIX \
 -t $TARGET_LANG_SUFFIX \
 --arch transformer \
 --share-all-embeddings \
 --optimizer adam \
 --adam-betas '(0.9, 0.98)' \
 --clip-norm 0.0 \
 --lr-scheduler inverse_sqrt \
 --warmup-init-lr 1e-07 \
 --warmup-updates 4000 \
 --lr 0.0005 \
 --min-lr 1e-09 \
 --dropout 0.3 \
 --weight-decay 0.0 \
 --criterion \
  label_smoothed_cross_entropy \
 --label-smoothing 0.1 \
 --max-tokens 4000 \
 --update-freq 8 \
 --save-dir $OUTPUT_FOLDER \
 --no-progress-bar \
 --log-interval 100 \
 --save-interval-updates 10000 \
 --keep-interval-updates 20 \
 --ddp-backend=no_c10d \
 --fp16
```

For finetuning, we change the following:

```
 --optimizer sgd \
 --lr-scheduler fixed \
 --lr 5e-5 \
```

D Segmentation Systems

The different hyperparameters values that were tried for the segmentation models are shown on Table 11. In total, no more than 75 combinations were tested in order to conduct the experiments reported on this paper.