Speech Translation with Style: AppTek's Submissions to the IWSLT Subtitling and Formality Tracks in 2023

Parnia Bahar[*], **Patrick Wilken**[*], **Javier Iranzo-Sánchez**,
Mattia di Gangi, Evgeny Matusov, Zoltán Tüske
Applications Technology (AppTek), Aachen, Germany
{pbahar,pwilken,jiranzo,mdigangi,ematusov,ztuske}@apptek.com

Abstract

AppTek participated in the subtitling and formality tracks of the IWSLT 2023 evaluation. This paper describes the details of our subtitling pipeline - speech segmentation, speech recognition, punctuation prediction and inverse text normalization, text machine translation and direct speech-to-text translation, intelligent line segmentation - and how we make use of the provided subtitling-specific data in training and fine-tuning. The evaluation results show that our final submissions are competitive, in particular outperforming the submissions by other participants by 5% absolute as measured by the SUBER subtitle quality metric. For the formality track, we participated with our En-Ru and En-Pt production models, which support formality control via prefix tokens. Except for informal Portuguese, we achieved near perfect formality level accuracy while at the same time offering high general translation quality.

1 Introduction

This paper presents AppTek's submissions to the subtitling and formality tracks of the IWSLT 2023 evaluation campaign. In the subtitling track, we participate in constrained and unconstrained conditions and in both language pairs English-to-German (En-De) and English-to-Spanish (En-Es). In the formality track, we participate in the zero-shot unconstrained condition for English-to-Portuguese (En-Pt) and English-to-Russian (En-Ru).

This paper is organized as follows: Section 2 briefly describes our data preparation. Section 3 presents AppTek's pipeline for subtitle translation. Its different components, namely audio segmentation, speech translation (ST), automatic speech recognition (ASR), machine translation (MT) models, and our subtitle segmentation algorithm are described in Sections 3.1-3.5. Section 3.6 contains experiments and an analysis of our subtitling systems. Section 4 presents AppTek's approach to formality-controlled machine translation. Finally, Section 4.1 shows the results of our formality track submission.

2 Data Preparation

2.1 Text Data

We use all of the allowed "speech-to-text parallel" and "text-parallel" data, including Europarl, Europarl-ST, News Commentary, CORDIS News, Tatoeba, TED2020, IWSLT TED, MuST-C v3, CoVoST v2, and OpenSubtitles[1]. We apply common parallel data filtering steps based on language identification, sentence length ratios between source and target sentences and additional heuristics. After filtering, we obtain 13.5M sentence pairs with 152M running words (counted on the English side) for En-De and 16.5M sentence pairs with 183M words for En-Es.

Next, we clone this data and process the En side of the clone with our text normalization tool NEWTN. It implements elaborate regular expressions to convert numbers, dates, monetary amounts, and other entities with digits into their spoken form. It is also used to remove punctuation and word case information. After training on such source data, our MT systems are able to directly translate from raw ASR output that lacks punctuation and casing into properly formatted written target language text.

For the parallel corpora which have document labels, we also create a version in which we concatenate two subsequent sentences from the same document using a separator symbol. Our past experience shows that adding such data is beneficial even if we do not add the context of the previous sentence at inference time.

Finally, for each language pair, we extract about 4M words of bilingual phrases (based on unsupervised word alignment) as additional training "sen-

[*]equal contribution

[1]The filtered version provided by the track organizers.

Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023), pages 251–260

tence" pairs to make sure that the MT system can cope well with incomplete sentences or too fine-grained automatic sentence segmentation.

2.2 Speech Data

We use all the allowed datasets marked as "speech" and "speech-to-text parallel", including Europarl-ST, How2, MuST-C, TED-LIUM, LibriSpeech, Mozilla Common Voice, VoxPopuli, CoVoST, and IWSLT TED. After removing very short (< 0.1s) and long (> 120s) segments, we obtain about 3590 hours of speech with transcripts. From each dataset, we only take the train sets, where applicable. The English text is processed to be lowercased, punctuation-free using NEWTN, and split into 10k byte-pair-encoding (BPE) tokens (Sennrich et al., 2016).

2.3 Direct Speech Translation Data

All data marked as "speech-to-text parallel", i.e. Europarl-ST, MuST-C, CoVoST, and IWSLT TED – except MuST-Cinema – is utilized for direct speech translation. It results in a total of approximately 1220 hours of speech with transcripts and corresponding translations after only keeping segments between 0.1 and 120 seconds. As for our data processing, on the English text, we carried out the same scheme as for speech data, while following almost the same German data processing scheme as described in Section 2.1. plus tokenization using the Moses toolkit (Koehn et al., 2007). Then 10k and 20k BPEs are used on the English and German texts, respectively. The dev set for the direct model is chosen to be the concatenation of IWSLT dev2010, MuST-C, Europal-ST, and CoVoST dev sets, resulting in a large dev set of 33 hours.

2.3.1 Synthetic Data

To leverage more training data for our direct model, we translate the English transcripts of the allowed "speech" data (Jia et al., 2019) using our constrained machine translation model described in Section 3.4 with output length control "short" (Wilken and Matusov, 2022). Combining the real ST data with the synthetic data, we obtain about 4100 hours of translated-speech parallel utterances.

3 Subtitle Translation

3.1 Audio Segmentation

We use the SHAS method (Tsiamas et al., 2022) for audio segmentation. SHAS scores every audio frame with a binary classifier (speech/no-speech), followed by a probabilistic divide-and-conquer (*pDAC*) algorithm that iteratively splits audio at the positions with the lowest probability of the speech class. For the unconstrained condition, we use the English segmentation model published by the authors of SHAS, which is an XLS-R 300M model (Babu et al., 2022) fine-tuned for the frame classification task on the MuST-C train set. For the constrained condition, we train our own frame classifier with Wav2Vec2 (Baevski et al., 2020), pre-trained on LibriSpeech, followed by fine-tuning for the frame classification task using MuST-C.

A hyper-parameter search was conducted to find the number of layers (constrained model), as well as the inference parameters (max. segment length and *pDAC* threshold) that optimize the performance of the downstream speech translation pipeline. We found that the *pDAC* threshold, which is the minimum probability required to keep a frame, has significant effects on the translation quality, and that the optimal value can vary depending on the task and acoustic conditions.

3.2 Direct Speech Translation

3.2.1 Attention Encoder-Decoder

We train an attention-based model (Bahdanau et al., 2015) composed of a Conformer encoder (Gulati et al., 2020) and a Transformer decoder (Vaswani et al., 2017). The encoder consists of 12 layers with a size of 512, a feed-forward size of 2048, and 8 heads, whereas the decoder has 6 layers with the same hidden size and number of heads. For fast yet stable convergence, we apply a layer-wise network construction scheme (Zeyer et al., 2018, 2019). Specifically, we start with 2 layers of halved hidden dimensions in both encoder and decoder (18M parameters) and linearly scale the model depth and width to full size (125M parameters) in the first 5 sub-epochs where each sub-epoch is one-twentieth of the whole training data. Also, L2-norm regularization and dropout are scaled up from 0 to 0.0001 and 0.1 respectively. Label smoothing is enabled only afterwards. We apply Adam (Kingma and Ba, 2015) with an initial learning rate of 0.0005 and dynamic learning scheduling based on dev set loss.

Audio log mel 80-dimensional features are extracted every 10ms. The first layer of Conformer is composed of 2 convolution layers with strides of 3 and 2 over time giving a reduction factor of 6. We use SpecAugment (Park et al., 2019; Bahar et al., 2019b) and speed perturbation in a random interval of $[0.9, 1.1]$ as data augmentation. In order

to train a single direct speech translation model that also supports time alignment between source label sequence and time frames, we add the source CTC loss (Graves et al., 2006; Kim et al., 2017; Bahar et al., 2019a) on top of the encoder in training.

We also add a second shallow 1-layer Transformer decoder (with 14M parameters) in order to generate better source transcripts for time alignment. Given this network with a shared speech encoder and two independent decoders, multi-task learning is employed to train all model parameters jointly. The final objective function is computed as a sum of the 3 losses (source CTC, source enc-dec, and target enc-dec).

3.2.2 Forced Alignment

CTC relies on Viterbi alignment to obtain the best path going through the source token at position n at time frame t. It is therefore possible to obtain word timings from CTC which can be used for subtitle generation. To do so, we first generate the source transcripts using the source decoder of the network and then use them to run forced-alignment on the CTC output. The model's alignments are on BPE-level, we therefore combine the timings of all subwords belonging to a word to obtain the final word-level timestamps.

We experimented with this approach and were able to generate accurate timestamps appropriate for creating subtitles in the source language. However, as we decide against using the source template approach for the constrained systems (see Section 3.5), only the timings of the first and last word in a segment are used for the target subtitles of the constrained submission. We plan to explore how to make better use of the CTC timings from this model in future experiments. In particular, we plan to add silence modeling to obtain information about pauses within speech segments, which can then be reflected in the subtitle timings.

3.3 Automatic Speech Recognition

Constrained We train a Conformer-Transformer model for the constrained task mainly following Section 3.2.1 using 3590 hours of speech. Layer-wise network construction, SpecAugment, and CTC loss are applied. Since the model is not trained for multiple tasks (no additional decoder is added), it has better performance in terms of WER compared to the source decoder part of the ST model. The final checkpoint achieves a WER of 9.6% on the concatenated dev set of 33h.

Unconstrained We train an attention-based encoder-decoder model to run ASR decoding and also a CTC model which is used to generate word timings by force-aligning the audio with the decoded hypotheses. Here, the CTC model uses an explicit word boundary `<space>` symbol between words. It serves as `silence` modeling. Both models are trained on the same training set of 15K hours of speech mixing publicly available data with a commercial license and in-house data.

The 185M-parameter attention-based model uses a 31-layer Conformer encoder of hidden size 384; 8 heads with 64 dimensions per head; Macaron-style (Lu et al., 2019) feed-forward layers with size 2048; convolutional layers with 1024 channels and kernel size 31. The decoder is a single-headed attention-based model (Tüske et al., 2020), and consists of 4 stacked projected long short-term memory (pLSTM) recurrent layers with layer size 2048 (Hochreiter and Schmidhuber, 1997; Sak et al., 2014). The first two LSTMs operate on the embedding of the label sequence only. The other two decoder LSTM layers also process the acoustic information extracted by the encoder using a single-head, additive, location-aware cross-attention. The decoder predicts 1K BPE units. Decoding is done using an external neural LM consisting of 4 stacked LSTM layers of size 3072 with the same output vocabulary as the ASR models. The 273M-parameter language model is trained on 2.4B running words segmented to BPE units. The language model data are selected from a wide range of various domains, e.g. books, movies, news, reviews, Wikipedia, talks, etc. ASR transcription is obtained after decoding with beam search limited to 16 hypotheses without any vocabulary constraints. The CTC model uses the same encoder structure as the attention-based model.

3.4 Machine Translation

3.4.1 Unconstrained Condition

For the unconstrained subtitling pipeline we use AppTek's production MT systems which have been trained on large amounts of parallel data, mostly from the OPUS collection (Tiedemann, 2012). Both En-De and En-Es systems are Transformer *Big* systems that support additional API parameters which can in particular control the genre (e.g. patents, news articles, dialogs) and length (automatic, short, long, etc.). The control is implemented via pseudo-tokens in the beginning of the source or target sentence (Matusov et al., 2020).

For the IWSLT experiments, we set the genre to "dialogs" because it reflects best the spoken spontaneous style in the dev 2023 data. When not mentioned otherwise, we set the length to "short". This yields more condensed translations, similar to how human subtitlers would translate to comply with a given reading speed limit.

3.4.2 Constrained Condition

For the constrained condition we use the parallel training data prepared as described in Section 2.1. As the dev data for learning rate control, we use the Europarl-ST and MuST-C dev sets.

Our MT model is a variant of the Transformer *Big* model (Vaswani et al., 2017) with additional encoder layers and using relative positional encoding (Shaw et al., 2018). We use a batch size of 800 words, but the effective batch size is increased by accumulating gradients over 8 batches. We add the same length control feature as for the unconstrained system by classifying the training data into 5 bins of target-to-source length ratios and adding the class label as a target-side prefix token.

We apply SentencePiece (Kudo and Richardson, 2018) segmentation with a vocabulary size of 10K for En and 20K for De/Es and use a translation factor to predict the casing of the target words (Wilken and Matusov, 2019). Our MT models have been trained for 100 sub-epochs with 1M lines in each; thus, all of the prepared data has been observed in training 1-3 times. For each sub-epoch, we select sentence pairs proportionally to the following distribution and then randomly mix them:

20% Europarl and Europarl-ST data
20% TED data (MuST-C, IWSLT, TED2020)
20% OpenSubtitles (other)
10% News (Commentary+CORDIS), Tatoeba, CoVoST
15% Concatenated neighboring sentence pairs[2]
5% OpenSubtitles (documentaries)
5% OpenSubtitles (sports)
5% Bilingual phrases

3.4.3 Length ROVER

For all final submissions, we optimize the length control of MT by using a length ROVER (Wilken and Matusov, 2022). For each segment we create 3 translations: without forcing the target-side length token, forcing length bin 2 ("short"), and forcing length bin 1 ("extra short"). From those translations we select the first – given the order above – that provides a translation with a target-to-source character ratio of less than 1.1. This is motivated by the fact that translations need to be fitted into the source subtitle template (Section 3.5.1). We note that the reading speed compliance of our submission could have been increased even further by exploiting timing information to select the MT length variants.

[2]See Section 2.1.

System	MuST-C	TED	EPTV	ITV	Peloton
English-to-German					
unconstrained	33.7	27.1	19.0	30.6	23.9
+ fine-tuning	35.0	27.7	20.3	31.0	24.4
constrained	32.3	34.2	18.4	27.2	20.3
+ fine-tuning	32.9	–	19.0	28.1	21.5
English-to-Spanish					
baseline	37.2	46.1	34.1	24.5	23.6
+ fine-tuning	38.2	46.4	34.8	25.5	24.7

Table 1: BLEU scores in % for text-only MT fine-tuning experiments on the MuST-C tst-COMMON set and on the AppTek's aligned subsets of the 2023 subtitling track dev data.

3.4.4 Fine-tuning Experiments

For our fine-tuning experiments, we first select "in-domain" training data in terms of similarity to the seed data – the dev 2023 set – from the real parallel data, as well as the synthetic data described in Section 2.3.1. The selection is done by clustering distributed sentence representations in the embedding space, and then keeping sentence pairs from the clusters which correspond to the seed data clusters. This is done considering both source and target seed data sentences, but independently, so that no sentence-level alignment of seed data is necessary. For details on this data selection method, please refer to our 2020 submission to the offline speech translation track (Bahar et al., 2020). With this method, we create two versions of the in-domain data: one using all 4 parts of the dev 2023 set as seed data (in-domain A: En-De: 1.9M lines, 27M En words; En-Es: 1.7M lines, 25M words), and one, for En-De only, using just ITV and Peloton dev 2023 parts as seed data (in-domain B: 1.5M lines, 20M words).

We then use the dev 2023 set as a dev set in fine-tuning of the MT model for learning rate control. Since the dev 2023 data is not aligned at sentence-level, but is available as (in part) independently created subtitle files, we had to sentence-align it. To do so, we first extracted full sentences from the English subtitles based on sentence-final punctuation marks, translated these sentences with the (constrained) baseline MT, and then re-

segmented the target side into sentences that match the source sentences using Levenshtein alignment as implemented by the SUBER tool (Wilken et al., 2022). The source-target segments obtained this way are kept in the final dev set only if the BERT F-score (Zhang et al., 2019) for a given pair is > 0.5 for TED, EPTV, and Peloton sets and > 0.55 for the ITV set. With this method, the obtained dev set contains 7645 sentence-like units with 27.7K words for TED, 2.3K for EPTV, 20.7K for Peloton, and 13.9K for ITV.

We perform fine-tuning for up to 20 sub-epochs ranging in size from 100K to 400K sentence pairs using a small learning rate between 10^{-06} and 10^{-05}, and select the best configuration for each of the four dev 2023 domains.

The fine-tuning results are shown in Table 1. Despite the fact that no real in-domain data, not even the dev 2023 set, is used as training data in fine-tuning we are able to improve MT quality in terms of BLEU scores (Papineni et al., 2002; Post, 2018), as well as BERT and other scores skipped due to space constraints. The improvements are more pronounced for the constrained system, but the absolute scores are generally better with the unconstrained setup[3]. However, since the TED talk and Europarl domains are covered well in the data allowed for the constrained condition, the difference between our unconstrained and constrained system for the TED and EPTV domains is small. It is worth noting that for ITV and Peloton domains we could only improve MT quality by fine-tuning on the in-domain B set that did not include any TED-related data, and also not using any TED or EPTV dev data for learning rate control.

3.5 Subtitle Creation

3.5.1 Source Template Approach

To create subtitle files from translation hypotheses, the text has to be segmented into blocks with start/end time information. One challenge is to transfer timings extracted from the source speech to the target subtitles. An approach to generate timings that is also used in human subtitling workflows (Georgakopoulou, 2019), is to first create subtitles in the source language – a so-called subtitle template – and to keep the same subtitle blocks during translation. This creates a nice viewing experience, since subtitles appear on the screen only during the actual speech. However, the source template constraints might be sub-optimal in terms of target language reading speed.

We use the source template approach for the unconstrained submission. To create subtitles in the original language of the videos (English), we start with a timed word list provided by the ASR system. We train a 3-layer bidirectional LSTM model (hidden size 256, embedding dim 128) to jointly add basic punctuation marks (`.,!?`) and casing information to the word list. As training data, we use 14M English sentences from the Gigaword and OpenSubtitles corpora. The model operates on full words and has two softmax output layers, one with the four punctuation tokens and "no punctuation" as target classes (to be added after the word), the other one with lower-cased, capitalized, all-upper, and mixed-cased classes as targets.

In addition, we train an inverse text normalization model to convert spoken forms of numbers, dates, currencies, etc. into the proper written form. This model is a Transformer *Big* trained on data where the source data is processed using our text normalization tool NEWTN, see Section 2.1. Applying it to the transcriptions helps MT to produce proper digits also on the target side. This has a slight positive effect on automatic scores (0.8% SUBER for Peloton, only up to 0.4% for the other domains), but mainly helps subjectively perceived quality and also reduces the number of characters.

The resulting timed, punctuated, and cased word list is split into sentences using punctuation (`.!?`) and pauses between words longer than 3 seconds. Those are fed into a subtitle segmentation algorithm similar to the one described in (Matusov et al., 2019). Its core component is an LSTM segmentation model that is trained on English OpenSubtitles XML data, which includes subtitle block boundary information[4], to estimate the probability of a subtitle break after each word of a given input sentence. Within a beam search framework, this model is combined with hard subtitling constraints such as the character limit per line to create valid subtitles. Here, we adjust it for the creation of subtitles from timed words by including minimum and maximum subtitle duration as constraints, and not forcing any predefined number of subtitles.

After segmentation, we use the start time of the

[3]The BLEU score of the constrained system on the En-De TED part is higher because, as we found out shortly before submission, some of the dev 2023 TED talks were part of the allowed TED2020 training corpus. Hence, further fine-tuning did not help for this system on this set. The unconstrained system had not been trained on this corpus.

[4]https://opus.nlpl.eu/download.php?f=OpenSubtitles/v2018/xml/en.zip

first word and the end time of the last word in each subtitle block as the subtitle start and end time. The subtitle template defined this way is then translated using the fine-tuned MT system described in Section 3.4.4, employing the length ROVER (Section 3.4.3) to avoid long translations that do not fit the template. Sentences as defined above are used as translation units, note that they may span several subtitle blocks. To insert the translations back into the template, we again apply the subtitle segmentation algorithm, this time with the exact settings as in (Matusov et al., 2019).

3.5.2 Template-Free Approach

By definition, the source template approach is not desirable for direct speech translation without intermediate source text representation. Also, the constrained condition does not include English OpenSubtitles data with subtitle breaks. We hence fall back to a simpler subtitle creation approach for our constrained direct and cascade systems. We use the segments provided by the audio segmenter as translation units. For the cascade system, we translate the transcription of each segment with the fine-tuned constrained MT, also using the length ROVER (Section 3.4.3). End-of-line and end-of-block tokens are inserted into the translated text of each segment using the subtitle segmentation algorithm configured similarly to the case of template creation in the previous section but without duration-based constraints. Timestamps for the additional subtitle block boundaries are then created by linearly interpolating the audio segment timings according to character count ratios. Assuming the translation of an audio segment with start time T_{start} and end time T_{end} is split into N blocks with $c_1, ..., c_N$ characters, respectively, the start time of block n is set to $T_{\text{start}} + (T_{\text{end}} - T_{\text{start}}) \cdot \frac{\sum_{n'=1}^{n-1} c_{n'}}{\sum_{n'=1}^{N} c_{n'}}$. This method leads to reasonable timings in most cases but can create temporary time shifts between speech and subtitles inside long audio segments.

3.5.3 Subtitle Post-Processing

To all subtitles, we apply a final post-processing that splits rare cases of subtitles with more than 2 lines (same segmentation method as for template-free approach) and shifts subtitle end times to later in time if needed to comply with the maximum reading speed of 21 characters per second. The latter is only possible if there is a large enough gap after a given subtitle and will therefore not guarantee low enough reading speed in all cases.

system	TED	EPTV	Peloton	ITV
SHAS 0.31	21.1	14.9	12.1	15.6
SHAS 0.50	22.4	14.9	11.6	13.9
SHAS 0.71	20.8	14.6	10.8	10.7
ASR Segm.	19.8	14.8	11.3	13.5

Table 2: Impact of different segmentation schemes on the translation quality (BLEU in %).

3.6 Results

We first decide which audio segmentation to use based on dev set results using our final ASR and MT unconstrained systems. We set different *pDAC* thresholds for the unconstrained SHAS (0.31, 0.50, and 0.71) and compare them with an in-house segmenter optimized for ASR. The results in Table 2 show that a low threshold of 0.31 leads to better translations overall. There is however variation depending on the domain: it is 1.3 BLEU points worse than SHAS 0.50 on TED, but as good or up to 1.7 BLEU points better in all other domains. Results for ITV are highly sensitive to the threshold. We attribute this to the fact that in TV series speech is often mixed with music and other sounds and a lower threshold is required not to miss speech segments. Given these results, we use SHAS 0.31 as our segmenter for unconstrained experiments. For the constrained experiments, we use SHAS 0.31 everywhere except on TED with SHAS 0.50.

Table 3 compares the performance of the final constrained cascade (separate ASR + MT) and direct En-De subtitling systems as well as the unconstrained cascade system. All metrics are computed using the SUBER tool[5] (Wilken et al., 2022) directly on subtitle files. To calculate the BLEU and CHRF (Popović, 2015) metrics, it performs an alignment of hypothesis to reference sentences similar to (Matusov et al., 2005). On all metrics, the constrained cascade system outperforms our direct model. We observe imperfections in the direct model's output such as repetitions. This can be partially attributed to the fact that it has been trained jointly for 3 tasks leading to sub-optimal optimization for the final translation process. The lack of length control of our direct ST model is another reason for the gap between the two constrained systems. For the cascade systems, we find length control via the length ROVER to be crucial, giving consistent improvements of 4 to 5% points in SUBER compared to no length control at all. As seen in Table 3, the unconstrained system out-

[5] https://github.com/apptek/SubER

system	constr.	SUBER (↓)	BLEU	CHRF
TED				
cascade	yes	63.0	26.0	53.9
direct	yes	75.9	17.1	47.6
cascade	no	64.3	22.1	51.0
EPTV				
cascade	yes	78.7	13.5	45.2
direct	yes	85.1	10.9	42.6
cascade	no	75.8	14.8	44.1
Peloton				
cascade	yes	87.6	9.9	32.0
direct	yes	86.1	6.8	26.9
cascade	no	71.9	11.6	34.3
ITV				
cascade	yes	83.6	8.5	26.1
direct	yes	90.9	5.7	21.0
cascade	no	71.4	14.8	35.2

Table 3: En-De subtitle translation results in % (constrained and unconstrained setting) on the dev2023 sets.

Domain	SUBER (↓)	BLEU (↑)	CHRF (↑)
TED	48.8	37.8	61.8
EPTV	70.2	20.4	50.6
Peloton	79.0	12.2	36.2
ITV	82.1	9.2	26.8

Table 4: Subtitle translation results in % on the dev2023 sets for En-Es via the constrained cascade system.

performs both constrained systems except on the TED set. This is due to a data overlap, some TED talks present in the dev set have also been part of the constrained training data. To analyze the impact of the source template approach we re-create the subtitles of the unconstrained system using the template-free approach. We find that this deteriorates the SUBER scores for TED, Peloton and ITV by 0.7, 3.6 and 3.8% points, respectively, while actually giving better results for EPTV by 0.7%. In general, the results in Table 3 show a higher automatic subtitling quality for the TED domain, which represents the case of well recorded and prepared speech, but also show the need to focus research on harder conditions such as interviews and TV series. Table 4 contains the scores we are able to achieve for En-Es under constrained conditions. Also here, acceptable subtitle quality can only be reached for TED and EPTV content, but not for the more challenging Peloton and ITV content.

4 Formality Control

AppTek's production systems support formality or, as we call it, style control for selected language pairs (Matusov et al., 2020). This year, we decided to test these systems in the unconstrained condition of the IWSLT formality track for En-Pt and En-to-Ru. Each of these two systems is trained in a Transformer Big setup (Vaswani et al., 2017). The formality level is encoded with a pseudo-token in the beginning of each training source sentence with one of 3 values: formal, informal, no style. The system is trained on large public data from the OPUS collection (Tiedemann, 2012) that has been partitioned into the 3 style classes as follows.

First, we write a sequence of regular expressions for the target language (in this case, European Pt and Ru) which try to match sentences containing formal or informal features. Thus, for Russian, we try to match either the formal or informal second-person pronoun that corresponds to English "you", including their possessive forms. For Portuguese, we additionally match the forms of most common verbs which agree with the corresponding pronoun. The regex list for Russian is given in Table 5[6].

Each list of regular expressions uses standard regex syntax and makes either case-sensitive or insensitive matches. For each sentence pair from the parallel data, the regex list is processed from top to bottom. As soon as a match in the target sentence is found, the FORMAL or INFORMAL label is assigned to the sentence pair. The sentence pair is labeled with NO_STYLE if there is no match.

If document information is available and at least 5% of the document sentence pairs are labeled as formal/informal according to the regex rules (with no sentences labeled with the opposite class), then all of the sentence pairs in the document are assigned the corresponding label. Such data is useful to model stylistic traits which are not limited to the choice of second-person pronouns. Note that document annotations are available for some of the IWSLT data, including TED talks, OpenSubtitles (each subtitle file corresponds to a document), individual sessions of European Parliament, etc.

We further smooth the three style classes to ensure that e.g., sentences containing second-person pronouns can be translated well even when no style is specified at inference time. To this end, 5 to 8% of sentence pairs which had been assigned to one of the 3 style classes as described above are randomly re-assigned to one of the other two classes.

For En-Ru, the training data that had been partitioned into style classes in this way included about

[6]We released the En-Pt and En-Ru lists of regular expressions as part of our evaluation submission.

INFORMAL IGNORECASE \b(ты|теб[яе]|тобой|тво[йеёяю]|твоей|твоего|твоему|твоим|тво[ёе]м)\b
FORMAL IGNORECASE \b(вы|вами?|ваш[ае]?|вашей|вашего|вашему?|вашу|вас|вашим)\b

Table 5: The regular expressions used to partition En-Ru training data into formal, informal, and (in case of no match) "no style" classes.

language pair / requested style		BLEU [%]	COMET	M-Acc [%]
En-Pt	formal	34.6	0.6089	99
	informal	42.4	0.6776	64
En-Ru	formal	35.4	0.6165	99
	informal	33.3	0.6026	98

Table 6: Automatic evaluation results for AppTek's submission to the formality track of IWSLT 2023.

40M sentence pairs. At the time this model was trained in early 2022, the larger CCMatrix corpus (Schwenk et al., 2021) was not included. For En-Pt, we did use a filtered version of CCMatrix in training, so that the total number of parallel sentence pairs was 140M. The filtering of CCMatrix and other large crawled data included removing sentence pairs with low cross-lingual sentence embedding similarity as given by the LABSE scores (Feng et al., 2022). All of our parallel training data is also filtered based on sentence-level language identification scores and other heuristics.

When training the Transformer Big model, we balanced the contribution of formal, informal, and "no style" data by adding them in equal proportions (number of lines) to each sub-epoch.

4.1 Results

We did not perform any experiments, but just set the API parameter `style=formal` or `style=informal` and translated the evaluation data with the AppTek's production systems, trained as described above. The results in terms of automatic error metrics, as reported by the track organizers, are summarized in Table 6.

Among the 5 participants of the unconstrained condition, we obtain the best results for En-Ru in terms of BLEU and COMET (Rei et al., 2020), while producing the correct formality level for more than 98% of the sentences. The second-best competitor system obtains formality accuracy of 100%, but scores 1.7% absolute lower in BLEU for the formal and 0.9% BLEU absolute for the informal class.

For En-Pt, our system scores second in terms of automatic MT quality metrics and correctly produced the formal style for 99% of the sentences in the evaluation data. However, when the informal style was requested, our system could generate it in only 64% of the cases. We attribute this low score to the imperfect regular expressions we defined for informal Portuguese pronouns and corresponding verb forms, since some of them are ambiguous. However, we find it difficult to explain that e.g. the BLEU score of AppTek's "informal" MT output with respect to the informal reference is almost 8% absolute higher than for our "formal" output with respect to the formal reference. This may indicate that the human reference translation also has not always followed the requested style, the informal one in particular.

5 Conclusion

We described AppTek's submissions to the subtitling and formality tracks of the IWSLT 2023.

For the subtitling track, we obtained good results, outperforming the other two evaluation participants either with our constrained or unconstrained cascaded approach on all 4 domains. Part of this success is due to our subtitle creation process, in which we employ AppTek's intelligent line segmentation models. However, the results varied by domain, with the domain of movie subtitles posing the most challenges for ASR, and the domain of fitness-related videos (Peloton) being hardest for MT. Yet our biggest overall challenge, especially for the direct (end-to-end) submission was speech segmentation and creating sentence-like units, on real ITV movies in particular, in which there is music, background noise, and multiple speakers. In the future, we plan to improve this component of our speech translation technology. We also plan to include length control in our direct models which showed to be an important factor for those applications with time constraints.

Our formality track participation was a one-shot attempt at a zero-shot task that showed the competitiveness of the formality control that we have implemented in AppTek's production systems. However, our approach currently requires the creation of manual regular expression rules for partitioning the parallel training data into formality classes, and the participation in the IWSLT evaluation revealed some weaknesses of this approach for one of the involved target languages. In the future, we plan to further improve our approach, reducing or eliminating the need for writing rules.

References

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2022. XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale. In *Proc. Interspeech 2022*, pages 2278–2282.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Parnia Bahar, Tobias Bieschke, and Hermann Ney. 2019a. A comparative study on end-to-end speech to text translation. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 792–799, Sentosa, Singapore.

Parnia Bahar, Patrick Wilken, Tamer Alkhouli, Andreas Guta, Pavel Golik, Evgeny Matusov, and Christian Herold. 2020. Start-before-end and end-to-end: Neural speech translation by apptek and rwth aachen university. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 44–54.

Parnia Bahar, Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2019b. On using specaugment for end-to-end speech translation. In *International Workshop on Spoken Language Translation (IWSLT)*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Panayota Georgakopoulou. 2019. Template files:: The holy grail of subtitling. *Journal of Audiovisual Translation*, 2(2):137–160.

Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *International Conference on Machine Learning (ICML)*, volume 148, pages 369–376, Pittsburgh, PA, USA.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *21th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 5036–5040.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J. Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. 2019. Leveraging weakly supervised data to improve end-to-end speech-to-text translation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 7180–7184. IEEE.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 4835–4839, New Orleans, LA, USA.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-yan Liu. 2019. Understanding and improving transformer from a multi-particle dynamic system point of view. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*.

Evgeny Matusov, Gregor Leusch, Oliver Bender, and Hermann Ney. 2005. Evaluating machine translation output with automatic sentence segmentation. In *International Workshop on Spoken Language Translation*, pages 148–154, Pittsburgh, PA, USA.

Evgeny Matusov, Patrick Wilken, and Yota Georgakopoulou. 2019. Customizing neural machine translation for subtitling. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 82–93, Florence, Italy. Association for Computational Linguistics.

Evgeny Matusov, Patrick Wilken, and Christian Herold. 2020. Flexible customization of a single neural machine translation system with multi-dimensional metadata inputs. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 2: User Track)*, pages 204–216, Virtual. Association for Machine Translation in the Americas.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Haşim Sak, Andrew W. Senior, and Françoise Beaufays. 2014. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. 2022. SHAS: Approaching optimal Segmentation for End-to-End Speech Translation. In *Proc. Interspeech 2022*, pages 106–110.

Zoltán Tüske, George Saon, Kartik Audhkhasi, and Brian Kingsbury. 2020. Single headed attention based sequence-to-sequence model for state-of-the-art results on Switchboard. In *Interspeech*, pages 551–555, Shanghai, China.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Patrick Wilken, Panayota Georgakopoulou, and Evgeny Matusov. 2022. SubER - a metric for automatic evaluation of subtitle quality. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 1–10, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Patrick Wilken and Evgeny Matusov. 2019. Novel applications of factored neural machine translation. *arXiv preprint arXiv:1910.03912*.

Patrick Wilken and Evgeny Matusov. 2022. AppTek's submission to the IWSLT 2022 isometric spoken language translation task. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 369–378, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Albert Zeyer, Parnia Bahar, Kazuki Irie, Ralf Schlüter, and Hermann Ney. 2019. A comparison of transformer and lstm encoder decoder models for asr. In *IEEE Automatic Speech Recognition and Understanding Workshop*, pages 8–15, Sentosa, Singapore.

Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney. 2018. Improved training of end-to-end attention models for speech recognition. In *19th Annual Conf. Interspeech, Hyderabad, India, 2-6 Sep.*, pages 7–11.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.