Segmentation Strategies for Streaming Speech Translation

Vivek Kumar Rangarajan Sridhar, John Chen, Srinivas Bangalore
Andrej Ljolje, Rathinavelu Chengalvarayan
AT&T Labs - Research
180 Park Avenue, Florham Park, NJ 07932
vkumar,jchen,srini,alj,rathi@research.att.com

Abstract

The study presented in this work is a first effort at real-time speech translation of TED talks, a compendium of public talks with different speakers addressing a variety of topics. We address the goal of achieving a system that balances translation accuracy and latency. In order to improve ASR performance for our diverse data set, adaptation techniques such as constrained model adaptation and vocal tract length normalization are found to be useful. In order to improve machine translation (MT) performance, techniques that could be employed in real-time such as monotonic and partial translation retention are found to be of use. We also experiment with inserting text segmenters of various types between ASR and MT in a series of real-time translation experiments. Among other results, our experiments demonstrate that a good segmentation is useful, and a novel conjunction-based segmentation strategy improves translation quality nearly as much as other strategies such as comma-based segmentation. It was also found to be important to synchronize various pipeline components in order to minimize latency.

1 Introduction

The quality of automatic speech-to-text and speech-to-speech (S2S) translation has improved so significantly over the last several decades that such systems are now widely deployed and used by an increasing number of consumers. Under the hood, the individual components such as automatic speech recognition (ASR), machine translation (MT) and text-to-speech synthesis (TTS) that constitute a S2S system are still loosely coupled and typically trained on disparate data and domains. Nevertheless, the models as well as the pipeline have been optimized in several ways to achieve tasks such as high quality offline speech translation (Cohen, 2007; Kingsbury et al., 2011; Federico et al., 2011), on-demand web based speech and text translation, low-latency real-time translation (Wahlster, 2000; Hamon et al., 2009; Bangalore et al., 2012), etc. The design of a S2S translation system is highly dependent on the nature of the audio stimuli. For example, talks, lectures and audio broadcasts are typically long and require appropriate segmentation strategies to chunk the input signal to ensure high quality translation. In contrast, single utterance translation in several consumer applications (apps) are typically short and can be processed without the need for additional chunking. Another key parameter in designing a S2S translation system for any task is latency. In offline scenarios where high latencies are permitted, several adaptation strategies (speaker, language model, translation model), denser data structures (N-best lists, word sausages, lattices) and rescoring procedures can be utilized to improve the quality of end-to-end translation. On the other hand, real-time speech-to-text or speech-to-speech translation demand the best possible accuracy at low latencies such that communication is not hindered due to potential delay in processing.

In this work, we focus on the speech translation of talks. We investigate the tradeoff between accuracy and latency for both offline and real-time translation of talks. In both these scenarios, appropriate segmentation of the audio signal as well as the ASR hypothesis that is fed into machine translation is critical for maximizing the overall translation quality of the talk. Ideally, one would like to train the models on entire talks. However, such corpora are not available in large amounts. Hence, it is necessary to con-

Proceedings of NAACL-HLT 2013, pages 230–238,
Atlanta, Georgia, 9–14 June 2013. ©2013 Association for Computational Linguistics

form to appropriately sized segments that are similar to the sentence units used in training the language and translation models. We propose several non-linguistic and linguistic segmentation strategies for the segmentation of text (reference or ASR hypotheses) for machine translation. We address the problem of latency in real-time translation as a function of the segmentation strategy; i.e., we ask the question "what is the segmentation strategy that maximizes the number of segments while still maximizing translation accuracy?".

2 Related Work

Speech translation of European Parliamentary speeches has been addressed as part of the TC-STAR project (Vilar et al., 2005; Fügen et al., 2006). The project focused primarily on offline translation of speeches. Simultaneous translation of lectures and speeches has been addressed in (Hamon et al., 2009; Fügen et al., 2007). However, the work focused on a single speaker in a limited domain. Offline speech translation of TED[1] talks has been addressed through the IWSLT 2011 and 2012 evaluation tracks. The talks are from a variety of speakers with varying dialects and cover a range of topics. The study presented in this work is the first effort on real-time speech translation of TED talks. In comparison with previous work, we also present a systematic study of the accuracy versus latency tradeoff for both offline and real-time translation on the same dataset.

Various utterance segmentation strategies for offline machine translation of text and ASR output have been presented in (Cettolo and Federico, 2006; Rao et al., 2007; Matusov et al., 2007). The work in (Fügen et al., 2007; Fügen and Kolss, 2007) also examines the impact of segmentation on offline speech translation of talks. However, the real-time analysis in that work is presented only for speech recognition. In contrast with previous work, we tackle the latency issue in simultaneous translation of talks as a function of segmentation strategy and present some new linguistic and non-linguistic methodologies. We investigate the accuracy versus latency tradeoff across translation of reference text, utterance segmented speech recognition output and partial speech recognition hypotheses.

3 Problem Formulation

The basic problem of text translation can be formulated as follows. Given a source (French) sentence $\mathbf{f} = f_1^J = f_1, \cdots, f_J$, we aim to translate it into target (English) sentence $\hat{\mathbf{e}} = \hat{e}_1^I = \hat{e}_1, \cdots, \hat{e}_I$.

$$\hat{\mathbf{e}}(\mathbf{f}) = \arg\max_{\mathbf{e}} \Pr(\mathbf{e}|\mathbf{f}) \tag{1}$$

If, as in talks, the source text (reference or ASR hypothesis) is very long, i.e., J is large, we attempt to break down the source string into shorter sequences, $\mathbf{S} = s_1 \cdots s_k \cdots s_{Q_s}$, where each sequence $s_k = [f_{j_k} f_{j_k+1} \cdots f_{j_{(k+1)}-1}]$, $j_1 = 1, j_{Q_s+1} = J+1$. Let the translation of each foreign sequence s_k be denoted by $t_k = [e_{i_k} e_{i_k+1} \cdots e_{i_{(k+1)}-1}]$, $i_1 = 1, i_{Q_s+1} = I^{'} + 1$[2]. The segmented sequences can be translated using a variety of techniques such as independent chunk-wise translation or chunk-wise translation conditioned on history as shown in Eqs. 2 and 3, respectively. In Eq. 3, t_i^* denotes the best translation for source sequence s_i.

$$\hat{\hat{\mathbf{e}}}(\mathbf{f}) = \arg\max_{t_1} \Pr(t_1|s_1) \cdots \arg\max_{t_k} \Pr(t_k|s_k) \tag{2}$$

$$\hat{\hat{\mathbf{e}}}(\mathbf{f}) = \arg\max_{t_1} \Pr(t_1|s_1) \arg\max_{t_2} \Pr(t_2|s_2, s_1, t_1^*) \cdots \arg\max_{t_k} \Pr(t_k|s_1, \cdots, s_k, t_1^*, \cdots, t_{k-1}^*) \tag{3}$$

Typically, the hypothesis $\hat{\hat{e}}$ will be more accurate than $\hat{e}$ for long texts as the models approximating $\Pr(\mathbf{e}|\mathbf{f})$ are conventionally trained on short text segments. In Eqs. 2 and 3, the number of sequences Q_s is inversely proportional to the time it takes to generate partial target hypotheses. Our main focus in this work is to obtain a segmentation $\mathbf{S}$ such that the quality of translation is maximized with minimal latency. The above formulation for automatic speech recognition is very similar except that the foreign string $\check{\mathbf{f}} = \check{f}_1^J = \check{f}_1, \cdots, \check{f}_{\check{J}}$ is obtained by decoding the input speech signal.

[1] http://www.ted.com

[2] The segmented and unsegmented talk may not be equal in length, i.e., $I \neq I^{'}$

	Model	Language	Vocabulary	#words	#sents	Corpora
ASR	Acoustic Model	en	46899	2611144	148460	1119 TED talks
	Language Model	en	378915	3398460155	151923101	Europarl, WMT11 Gigaword, WMT11 News crawl WMT11 News-commentary, WMT11 UN, IWSLT11 TED training
MT	Parallel text	en	503765	76886659	7464857	IWSLT11 TED training talks, Europarl, JRC-ACQUIS Opensubtitles, Web data
		es	519354	83717810	7464857	
	Language Model	es	519354	83717810	7464857	Spanish side of parallel text

Table 1: Statistics of the data used for training the speech translation models.

4 Data

In this work, we focus on the speech translation of TED talks, a compendium of public talks from several speakers covering a variety of topics. Over the past couple of years, the International Workshop on Spoken Language Translation (IWSLT) has been conducting the evaluation of speech translation on TED talks for English-French. We leverage the IWSLT TED campaign by using identical development (dev2010) and test data (tst2010). However, English-Spanish is our target language pair as our internal projects are cater mostly to this pair. As a result, we created parallel text for English-Spanish based on the reference English segments released as part of the evaluation (Cettolo et al., 2012).

We also harvested the audio data from the TED website for building an acoustic model. A total of 1308 talks in English were downloaded, out of which we used 1119 talks recorded prior to December 2011. We split the stereo audio file and duplicated the data to account for any variations in the channels. The data for the language models was also restricted to that permitted in the IWSLT 2011 evaluation. The parallel text for building the English-Spanish translation model was obtained from several corpora: Europarl (Koehn, 2005), JRC-Acquis corpus (Steinberger et al., 2006), Opensubtitle corpus (Tiedemann and Lars Nygaard, 2004), Web crawling (Rangarajan Sridhar et al., 2011) as well as human translation of proprietary data. Table 1 summarizes the data used in building the models. It is important to note that the IWSLT evaluation on TED talks is completely offline. In this work, we perform the first investigation into the real-time translation of these talks.

5 Speech Translation Models

In this section, we describe the acoustic, language and translation models used in our experiments.

5.1 Acoustic and Language Model

We use the AT&T WATSON^{SM} speech recognizer (Goffin et al., 2004). The speech recognition component consisted of a three-pass decoding approach utilizing two acoustic models. The models used three-state left-to-right HMMs representing just over 100 phonemes. The phonemes represented general English, spelled letters and head-body-tail representation for the eleven digits (with ”zero” and ”oh”). The pronunciation dictionary used the appropriate phoneme subset, depending on the type of the word. The models had 10.5k states and 27k HMMs, trained on just over 300k utterances, using both of the stereo channels. The baseline model training was initialized with several iterations of ML training, including two builds of context dependency trees, followed by three iterations of Minimum Phone Error (MPE) training.

The Vocal Tract Length Normalization (VTLN) was applied in two different ways. One was estimated on an utterance level, and the other at the talk level. No speaker clustering was attempted in training. The performance at test time was comparable for both approaches on the development set. Once the warps were estimated, after five iterations, the ML trained model was updated using MPE training. Constrained model adaptation (CMA) was applied to the warped features and the adapted features were recognized in the final pass with the VTLN model. All the passes used the same LM. For offline recognition the warps, and the CMA adaptation, are performed at the talk level. For the real-time speech translation experiments, we used the VTLN model.

The English language model was built using the permissible data in the IWSLT 2011 evaluation. The texts were normalized using a variety of cleanup, number and spelling normalization techniques and filtered by restricting the vocabulary to the top 375000 types; i.e., any sentence containing a token outside the vocabulary was discarded. First, we removed extraneous characters beyond the ASCII range followed by removal of punctuations. Subsequently, we normalized hyphenated words and removed words with more than 25 characters. The resultant text was normalized using a variety of number conversion routines and each corpus was filtered by restricting the vocabulary to the top 150000 types; i.e., any sentence containing a token outside the vocabulary was discarded. The vocabulary from all the corpora was then consolidated and another round of filtering to the top 375000 most frequent types was performed. The OOV rate on the TED dev2010 set is 1.1%. We used the AT&T FSM toolkit (Mohri et al., 1997) to train a trigram language model (LM) for each component (corpus). Finally, the component language models were interpolated by minimizing the perplexity on the dev2010 set. The results are shown in Table 2.

	Accuracy (%)	
Model	dev2010	test2010
Baseline MPE	75.5	73.8
VTLN	78.8	77.4
CMA	80.5	80.0

Table 2: ASR word accuracies on the IWSLT data sets.[3]

5.2 Translation Model

We used the Moses toolkit (Koehn et al., 2007) for performing statistical machine translation. Minimum error rate training (MERT) was performed on the development set (dev2010) to optimize the feature weights of the log-linear model used in translation. During decoding, the unknown words were preserved in the hypotheses. The data used to train the model is summarized in Table 1.

We also used a finite-state implementation of translation without reordering. Reordering can pose a challenge in real-time S2S translation as the text-to-speech synthesis is monotonic and cannot retract already synthesized speech. While we do not address the text-to-speech synthesis of target text in this work, we perform this analysis as a precursor to future work. We represent the phrase translation table as a weighted finite state transducer (FST) and the language model as a finite state acceptor (FSA). The weight on the arcs of the FST is the dot product of the MERT weights with the translation scores. In addition, a word insertion penalty was also applied to each word to penalize short hypotheses. The decoding process consists of composing all possible segmentations of an input sentence with the phrase table FST and language model, followed by searching for the best path. Our FST-based translation is the equivalent of phrase-based translation in Moses without reordering. We present results using the independent chunk-wise strategy and chunk-wise translation conditioned on history in Table 3. The chunk-wise translation conditioned on history was performed using the *continue-partial-translation* option in Moses.

6 Segmentation Strategies

The output of ASR for talks is a long string of words with no punctuation, capitalization or segmentation markers. In most offline ASR systems, the talk is first segmented into short utterance-like audio segments before passing them to the decoder. Prior work has shown that additional segmentation of ASR hypotheses of these segments may be necessary to improve translation quality (Rao et al., 2007; Matusov et al., 2007). In a simultaneous speech translation system, one can neither find the optimal segmentation of the entire talk nor tolerate high latencies associated with long segments. Consequently, it is necessary to decode the incoming audio incrementally as well as segment the ASR hypotheses appropriately to maximize MT quality. We present a variety of linguistic and non-linguistic segmentation strategies for segmenting the source text input into MT. In our experiments, they are applied to different inputs including reference text, ASR 1-best hypothesis for manually segmented audio and

[3] We used the standard NIST scoring package as we did not have access to the IWSLT evaluation server that may normalize and score differently

incremental ASR hypotheses from entire talks.

6.1 Non-linguistic segmentation

The simplest method is to segment the incoming text according to length in number of words. Such a procedure can destroy semantic context but has little to no overhead in additional processing. We experiment with segmenting the text according to word window sizes of length 4, 8, 11, and 15 (denoted as data sets *win4*, *win8*, *win11*, *win15*, respectively in Table 3). We also experiment with concatenating all of the text from one TED talk into a single chunk (*complete talk*).

A novel hold-output model was also developed in order to segment the input text. Given a pair of parallel sentences, the model segments the source sentence into minimally sized chunks such that crossing links and links of one target word to many source words in an optimal GIZA++ alignment (Och and Ney, 2003) occur only within individual chunks. The motivation behind this model is that if a segment s_0 is input at time t_0 to an incremental MT system, it can be translated right away without waiting for a segment s_i that is input at a later time $t_i, t_i > 0$. The hold-output model detects these kinds of segments given a sequence of English words that are input from left to right. A kernel-based SVM was used to develop this model. It tags a token t in the input with either the label HOLD, meaning to chunk it with the next token, or the label OUTPUT, meaning to output the chunk constructed from the maximal consecutive sequence of tokens preceding t that were all tagged as HOLD. The model considers a five word and POS window around the target token t. Unigram, bigram, and trigram word and POS features based upon this window are used for classification. Training and development data for the model was derived from the English-Spanish TED data (see Table 1) after running it through GIZA++. Accuracy of the model on the development set was 66.62% F-measure for the HOLD label and 82.75% for the OUTPUT label.

6.2 Linguistic segmentation

Since MT models are trained on parallel text sentences, we investigate segmenting the source text into sentences. We also investigate segmenting the text further by predicting comma separated chunks within sentences. These tasks are performed by training a kernel-based SVM (Haffner et al., 2003) on a subset of English TED data. This dataset contained 1029 human-transcribed talks consisting of about 103,000 sentences containing about 1.6 million words. Punctuation in this dataset was normalized as follows. Different kinds of sentence ending punctuations were transformed into a uniform end of sentence marker. Double-hyphens were transformed into commas. Commas already existing in the input were kept while all other kinds of punctuation symbols were deleted. A part of speech (POS) tagger was applied to this input. For speed, a unigram POS tagger was implemented which was trained on the Penn Treebank (Marcus et al., 1993) and used orthographic features to predict the POS of unknown words. The SVM-based punctuation classifier relies on a five word and POS window in order to classify the target word. Specifically, token t_0 is classified given as input the window $t_{-2}t_{-1}t_ot_1t_2$. Unigram, bigram, and trigram word and POS features based on this window were used for classification. Accuracy of the classifier on the development set was 60.51% F-measure for sentence end detection and 43.43% F-measure for comma detection. Subsequently, data sets *pred-sent* (sentences) and *pred-punct* (comma-separated chunks) were obtained. Corresponding to these, two other data sets *ref-sent* and *ref-punct* were obtained based upon gold-standard punctuations in the reference.

Besides investigating the use of comma-separated segments, we investigated other linguistically motivated segments. These included conjunction-word based segments. These segments are separated at either conjunction (e.g. "and," "or") or sentence-ending word boundaries. Conjunctions were identified using the unigram POS tagger. F-measure performance for detecting conjunctions by the tagger on the development set was quite high, 99.35%. As an alternative, text chunking was performed within each sentence, with each chunk corresponding to one segment. Text chunks are non-recursive syntactic phrases in the input text. We investigated segmenting the source into text chunks using TreeTagger, a decision-tree based text chunker (Schmid, 1994). Initial sets of text chunks were created by using either gold-standard sentence boundaries or boundaries detected using the punctuation classifier, yielding the data sets *chunk-ref-*

		Reference text				ASR 1-best			
		BLEU			Mean	BLEU			Mean
Segmentation	Segmentation	Independent chunk-wise		chunk-wise	#words	Independent chunk-wise		chunk-wise	#words
type	strategy	FST	Moses	with history	per segment	FST	Moses	with history	per segment
Non-linguistic	win4	22.6	21.0	25.5	3.9±0.1	17.7	17.1	20.0	3.9±0.1
	win8	26.6	26.2	28.2	7.9±0.3	20.6	20.9	22.3	7.9±0.2
	win11	27.2	27.4	29.2	10.9 ± 0.3	21.5	21.8	23.1	10.9±0.4
	win15	28.5	28.5	29.4	14.9±0.6	22.3	22.8	23.3	14.9±0.7
	ref-hold	13.3	14.0	17.1	1.6±1.9	12.7	13.1	17.5	1.5±1.0
	pred-hold	15.9	15.7	16.3	2.2±1.9	12.6	12.9	17.4	1.5±1.0
	complete talk	23.8	23.9	–	2504	18.8	19.2	–	2515
Linguistic	ref-sent	30.6	31.5	30.5	16.7±11.8	24.3	25.1	24.4	17.0±11.6
	ref-punct	30.4	31.5	30.3	7.1±5.3	24.2	25.1	24.1	8.7±6.1
	pred-punct	30.6	31.5	30.4	8.7±8.8	24.1	25.0	24.0	8.8±6.8
	conj-ref-eos	30.5	31.5	30.2	11.2±7.5	24.1	24.9	24.0	11.5±7.7
	conj-pred-eos	30.3	31.2	30.3	10.9±7.9	24.0	24.8	24.0	11.4±8.5
	chunk-ref-punct	17.9	18.9	21.4	1.3±0.7	14.5	15.2	16.9	1.4±0.7
	lgchunk1-ref-punct	21.0	21.8	25.1	1.7±1.0	16.9	17.4	19.6	1.8±1.0
	lgchunk2-ref-punct	22.4	23.1	26.0	2.1±1.1	17.9	18.4	20.4	2.1±1.1
	lgchunk3-ref-punct	24.3	25.1	27.4	2.5±1.7	19.2	19.9	21.3	2.5±1.7
	chunk-pred-punct	17.9	18.9	21.4	1.3±0.7	14.5	15.1	16.9	1.4±0.7
	lgchunk1-pred-punct	21.2	21.9	25.2	1.8±1.0	16.7	17.2	19.7	1.8±1.0
	lgchunk2-pred-punct	22.6	23.1	26.0	2.1±1.2	17.7	18.3	20.5	2.1±1.2
	lgchunk3-pred-punct	24.5	25.3	27.4	2.6±1.8	19.1	20.0	21.3	2.5±1.7

Table 3: BLEU scores at the talk level for reference text and ASR 1-best for various segmentation strategies. The ASR 1-best was performed on manually segmented audio chunks provided in *tst2010* set.

punct and *chunk-pred-punct*. Chunk types included NC (noun chunk), VC (verb chunk), PRT (particle), and ADVC (adverbial chunk).

Because these chunks may not provide sufficient context for translation, we also experimented with concatenating neighboring chunks of certain types to form larger chunks. Data sets *lgchunk1* concatenate together neighboring chunk sequences of the form NC, VC or NC, ADVC, VC, intended to capture as single chunks instances of subject and verb. In addition to this, data sets *lgchunk2* capture chunks such as PC (prepositional phrase) and VC followed by VC (control and raising verbs). Finally, data sets *lgchunk3* capture as single chunks VC followed by NC and optionally followed by PRT (verb and its direct object).

Applying the conjunction segmenter after the aforementioned punctuation classifier in order to detect the ends of sentences yields the data set *conj-pred-eos*. Applying it on sentences derived from the gold-standard punctuations yields the data set *conj-ref-eos*. Finally, applying the hold-output model to sentences derived using the punctuation classifier produces the data set *pred-hold*. Obtaining English sentences tagged with HOLD and OUTPUT directly from the output of GIZA++ on English-Spanish sentences in the reference produces the data set *ref-hold*. The strategies containing the keyword *ref* for ASR simply means that the ASR hypotheses are used in place of the gold reference text.

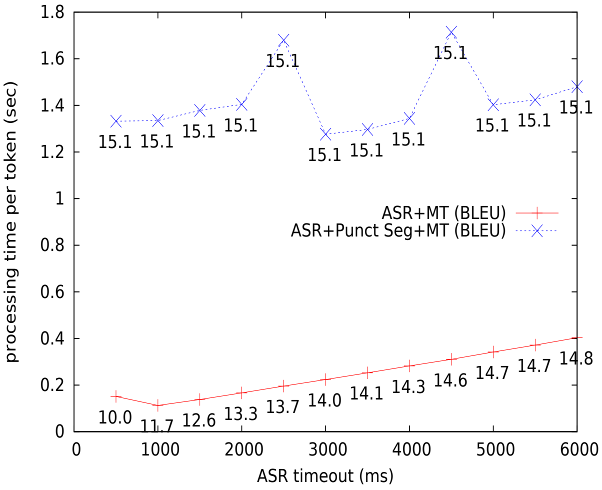

Figure 1: Latencies and BLEU scores for tst2010 set using incremental ASR decoding and translation

We also performed real-time speech translation by using incremental speech recognition, i.e., the decoder returns partial hypotheses that, independent of the pruning during search, will not change in the future. Figure 1 shows the plot for two scenarios: one in which the partial hypotheses are sent directly to machine translation and another where the best segmentation strategy *pred-punct* is used to segment the partial output before sending it to MT. The plot shows the BLEU scores as a function of ASR timeouts used to generate the partial hypotheses. Figure 1 also shows the average latency involved in incremental speech translation.

7 Discussion

The BLEU scores for the segmentation strategies over ASR hypotheses was computed at the talk level. Since the ASR hypotheses do not align with the reference source text, it is not feasible to evaluate the translation performance using the gold reference. While other studies have used an approximate edit distance algorithm for resegmentation of the hypotheses (Matusov et al., 2005), we simply concatenate all the segments and perform the evaluation at the talk level.

The *hold* segmentation strategy yields the poorest translation performance. The significant drop in BLEU score can be attributed to relatively short segments (2-4 words) that was generated by the model. The scheme oversegments the text and since the translation and language models are trained on sentence like chunks, the performance is poor. For example, the input text *the sea* should be translated as *el mar*, but instead the *hold* segmenter chunks it as *the·sea* which MT's chunk translation renders as *el·el mar*. It will be interesting to increase the span of the *hold* strategy to subsume more contiguous sequences and we plan to investigate this as part of future work.

The *chunk* segmentation strategy yields quite poor translation performance. In general, it does not make the same kinds of errors that the *hold* strategy makes; for example, the input text *the sea* will be treated as one NC chunk by the *chunk* segmentation strategy, leading MT to translate it correctly as *el mar*. The short chunk sizes of *chunk* lead to other kinds of errors. For example, the input text *we use* will be chunked into the NC *we* and the VC *use*, which will be translated incorrectly as *nosotros·usar*; the infinitive *usar* is selected rather than the properly conjugated form *usamos*. However, there is a marked improvement in translation accuracy with increasingly larger chunk sizes (*lgchunk1*, *lgchunk2*, and *lgchunk3*). Notably, *lgchunk3* yields performance that approaches that of *win8* with a chunk size that is one third of *win8*'s.

The *conj-pred-eos* and *pred-punct* strategies work the best, and it can be seen that the average segment length (8-12 words) generated in both these schemes is very similar to that used for training the models. It is also about the average latency (4-5 seconds) that can be tolerated in cross-lingual communication, also known as ear-voice span (Lederer, 1978). The non-linguistic segmentation using fixed word length windows also performs well, especially for the longer length windows. However, longer windows (*win15*) increase the latency and any fixed length window typically destroys the semantic context. It can also be seen from Table 3 that translating the complete talk is suboptimal in comparison with segmenting the text. This is primarily due to bias on sentence length distributions in the training data. Training models on complete talks is likely to resolve this issue. Contrasting the use of reference segments as input to MT (*ref-sent*, *ref-punct*, *conj-ref-eos*) versus the use of predicted segments (*pred-sent*, *pred-punct*, *conj-pred-eos*, respectively), it is interesting to note that the MT accuracies never differed greatly between the two, despite the noise in the set of predicted segments.

The performance of the real-time speech translation of TED talks is much lower than the offline scenario. First, we use only a VTLN model as performing CMA adaptation in a real-time scenario typically increases latency. Second, the ASR language model is trained on sentence-like units and decoding the entire talk with this LM is not optimal. A language model trained on complete talks will be more appropriate for such a framework and we are investigating this as part of current work.

Comparing the accuracies of different speech translation strategies, Table 3 shows that *pred-punct* performs the best. When embedded in an incremental MT speech recognition system, Figure 1 shows that it is more accurate than the system that sends partial ASR hypotheses directly to MT. This advantage decreases, however, when the ASR timeout parameter is increased to more than five or six seconds. In terms of latency, Figure 1 shows that the addition of the *pred-punct* segmenter into the incremental system introduces a significant delay. About one third of the increase in delay can be attributed to merely maintaining the two word lookahead window that the segmenter's classifier needs to make decisions. This is significant because this kind of window has been used quite frequently in previous work on simultaneous translation (cf. (Fügen et al., 2007)), and yet to our knowledge this penalty associated with this configuration was never mentioned. The remaining delay can be attributed to the long chunk sizes that the segmenter produces. An interesting aspect of the latency curve associated with the segmenter in Figure 1 is that there are two peaks at ASR timeouts of 2,500 and 4,500 ms, and that the lowest latency is achieved at 3,000 ms rather than at a smaller value. This may be attributed to the fact that the system is a pipeline consisting of ASR, segmenter, and MT, and that 3,000 ms is roughly the length of time to recite comma-separated chunks. Consequently, the two latency peaks appear to correspond with ASR producing segments that are most divergent with segments that the segmenter produces, leading to the most pipeline "stalls." Conversely, the lowest latency occurs when the timeout is set so that ASR's segments most resemble the segmenter's output to MT.

8 Conclusion

We investigated various approaches for incremental speech translation of TED talks, with the aim of producing a system with high MT accuracy and low latency. For acoustic modeling, we found that VTLN and CMA adaptation were useful for increasing the accuracy of ASR, leading to a word accuracy of 80% on TED talks used in the IWSLT evaluation track. In our offline MT experiments retention of partial translations was found useful for increasing MT accuracy, with the latter being slightly more helpful. We experimented with several linguistic and non-linguistic strategies for text segmentation before translation. Our experiments indicate that a novel segmentation into conjunction-separated sentence chunks resulted in accuracies almost as high and latencies almost as short as comma-separated sentence chunks. They also indicated that significant noise in the detection of sentences and punctuation did not seriously impact the resulting MT accuracy. Experiments on real-time simultaneous speech translation using partial recognition hypotheses demonstrate that introduction of a segmenter increases MT accuracy. They also showed that in order to reduce latency it is important for buffers in different pipeline components to be synchronized so as to minimize pipeline stalls. As part of future work, we plan to extend the framework presented in this work for performing speech-to-speech translation. We also plan to address the challenges involved in S2S translation across languages with very different word order.

Acknowledgments

We would like to thank Simon Byers for his help with organizing the TED talks data.

References

S. Bangalore, V. K. Rangarajan Sridhar, P. Kolan, L. Golipour, and A. Jimenez. 2012. Real-time incremental speech-to-speech translation of dialogs. In *Proceedings of NAACL:HLT*, June.

M. Cettolo and M. Federico. 2006. Text segmentation criteria for statistical machine translation. In *Proceedings of the 5th international conference on Advances in Natural Language Processing*.

M. Cettolo, C. Girardi, and M. Federico. 2012. WIT3: Web Inventory of Transcribed and Translated Talks. In *Proceedings of EAMT*.

J. Cohen. 2007. The GALE project: A description and an update. In *Proceedings of ASRU Workshop*.

M. Federico, L. Bentivogli, M. Paul, and S. Stüker. 2011. Overview of the IWSLT 2011 evaluation campaign. In *Proceedings of IWSLT*.

C. Fügen and M. Kolss. 2007. The influence of utterance chunking on machine translation performance. In *Proceedings of Interspeech*.

C. Fügen, M. Kolss, D. Bernreuther, M. Paulik, S. Stuker, S. Vogel, and A. Waibel. 2006. Open domain speech recognition & translation: Lectures and speeches. In *Proceedings of ICASSP*.

C. Fügen, A. Waibel, and M. Kolss. 2007. Simultaneous translation of lectures and speeches. *Machine Translation*, 21:209–252.

V. Goffin, C. Allauzen, E. Bocchieri, D. Hakkani-Tür, A. Ljolje, and S. Parthasarathy. 2004. The AT&T Watson Speech Recognizer. Technical report, September.

P. Haffner, G. Tür, and J. Wright. 2003. Optimizing svms for complex call classification. In *Proceedings of ICASSP'03*.

O. Hamon, C. Fügen, D. Mostefa, V. Arranz, M. Kolss, A. Waibel, and K. Choukri. 2009. End-to-end evaluation in simultaneous translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, March.

B. Kingsbury, H. Soltau, G. Saon, S. Chu, Hong-Kwang Kuo, L. Mangu, S. Ravuri, N. Morgan, and A. Janin. 2011. The IBM 2009 GALE Arabic speech translation system. In *Proceedings of ICASSP*.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, Shen W., C. Moran, R. Zens, C. J. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL*.

P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.

M. Lederer. 1978. Simultaneous interpretation: units of meaning and other features. In D. Gerver and H. W. Sinaiko, editors, *Language interpretation and communication*, pages 323–332. Plenum Press, New York.

M. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2):313–330.

E. Matusov, G. Leusch, O. Bender, and H. Ney. 2005. Evaluating machine translation output with automatic sentence segmentation. In *Proceedings of IWSLT*.

E. Matusov, D. Hillard, M. Magimai-Doss, D. Hakkani-Tür, M. Ostendorf, and H. Ney. 2007. Improving speech translation with automatic boundary prediction. In *Proceedings of Interspeech*.

M. Mohri, F. Pereira, and M. Riley. 1997. At&t general-purpose finite-state machine software tools, http://www.research.att.com/sw/tools/fsm/.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

V. K. Rangarajan Sridhar, L. Barbosa, and S. Bangalore. 2011. A scalable approach to building a parallel corpus from the Web. In *Proceedings of Interspeech*.

S. Rao, I. Lane, and T. Schultz. 2007. Optimizing sentence segmentation for spoken language translation. In *Proceedings of Interspeech*.

H. Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*.

R. Steinberger, B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, and D. Tufis. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of LREC*.

J. Tiedemann and L. Lars Nygaard. 2004. The OPUS corpus - parallel & free. In *Proceedings of LREC*.

D. Vilar, E. Matusov, S. Hasan, R. Zens, and H. Ney. 2005. Statistical machine translation of European parliamentary speeches. In *Proceedings of MT Summit*.

W. Wahlster, editor. 2000. *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer.