Real-time Incremental Speech-to-Speech Translation of Dialogs

Srinivas Bangalore, Vivek Kumar Rangarajan Sridhar, Prakash Kolan
Ladan Golipour, Aura Jimenez
AT&T Labs - Research
180 Park Avenue
Florham Park, NJ 07932, USA
vkumar,srini,pkolan,ladan,aura@research.att.com

Abstract

In a conventional telephone conversation between two speakers of the same language, the interaction is real-time and the speakers process the information stream incrementally. In this work, we address the problem of incremental speech-to-speech translation (S2S) that enables cross-lingual communication between two remote participants over a telephone. We investigate the problem in a novel real-time Session Initiation Protocol (SIP) based S2S framework. The speech translation is performed incrementally based on generation of partial hypotheses from speech recognition. We describe the statistical models comprising the S2S system and the SIP architecture for enabling real-time two-way cross-lingual dialog. We present dialog experiments performed in this framework and study the tradeoff in accuracy versus latency in incremental speech translation. Experimental results demonstrate that high quality translations can be generated with the incremental approach with approximately half the latency associated with non-incremental approach.

1 Introduction

In recent years, speech-to-speech translation (S2S) technology has played an increasingly important role in narrowing the language barrier in cross-lingual interpersonal communication. The improvements in automatic speech recognition (ASR), statistical machine translation (MT), and, text-to-speech synthesis (TTS) technology has facilitated the serial binding of these individual components to achieve S2S translation of acceptable quality.

Prior work on S2S translation has primarily focused on providing either one-way or two-way translation on a single device (Waibel et al., 2003; Zhou et al., 2003). Typically, the user interface requires the participant(s) to choose the source and target language apriori. The nature of communication, either single user talking or turn taking between two users can result in a one-way or cross-lingual dialog interaction. In most systems, the necessity to choose the directionality of translation for each turn does take away from a natural dialog flow. Furthermore, single interface based S2S translation (embedded or cloud-based) is not suitable for cross-lingual communication when participants are geographically distant, a scenario more likely in a global setting. In such a scenario, it is imperative to provide real-time and low latency communication.

In a conventional telephone conversation between two speakers of the same language, the interaction is real-time and the speakers process the information stream incrementally. Similarly, cross-lingual dialog between two remote participants will greatly benefit through incremental translation. While incremental decoding for text translation has been addressed previously in (Furuse and Iida, 1996; Sankaran et al., 2010), we address the problem in a speech-to-speech translation setting for enabling real-time cross-lingual dialog. We address the problem of incrementality in a novel session initiation protocol (SIP) based S2S translation system that enables two people to interact and engage in cross-lingual dialog over a telephone (mobile phone or landline). Our system performs incremental speech recognition and translation, allowing for low latency interaction that provides an ideal setting for remote dialog aimed at accomplishing a task.

We present previous work in this area in Section 2 and introduce the problem of incremental translation in Section 3. We describe the statistical models used in the S2S translation framework in Section 4 followed by a description of the SIP communication

2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 437–445,
Montréal, Canada, June 3-8, 2012. ©2012 Association for Computational Linguistics

framework for real-time translation in Section 5. In Section 6, we describe the basic call flow of our system following which we present dialog experiments performed using our framework in Section 8. Finally, we conclude in Section 9 along with directions for future work.

2 Previous Work

Most previous work on speech-to-speech translation systems has focused on a single device model, i.e., the user interface for translation is on one device (Waibel et al., 1991; Metze et al., 2002; Zhou et al., 2003; Waibel et al., 2003). The device typically supports multiple source-target language pairs. A user typically chooses the directionality of translation and a toggle feature is used to switch the directionality. However, this requires physical presence of the two conversants in one location.

On the other hand, text chat between users over cell phones has become increasingly popular in the last decade. While the language used in the interaction is typically monolingual, there have been attempts to use statistical machine translation to enable cross-lingual text communication (Chen and Raman, 2008). But this introduces a significant overhead as the users need to type in the responses for each turn. Moreover, statistical translation systems are typically unable to cope with telegraphic text present in chat messages. A more user friendly approach would be to use speech as the modality for communication.

One of the first attempts for two-way S2S translation over a telephone between two potentially remote participants was made as part of the Verbmobil project (Wahlster, 2000). The system was restricted to certain topics and speech was the only modality. Furthermore, the spontaneous translation of dialogs was not incremental. One of the first attempts at incremental text translation was demonstrated in (Furuse and Iida, 1996) using a transfer-driven machine translation approach. More recently, an incremental decoding framework for text translation was presented in (Sankaran et al., 2010). To the best of our knowledge, incremental speech-to-speech translation in a dialog setting has not been addressed in prior work. In this work, we address this problem using first of a kind SIP-based large vocabulary S2S translation system that can work with both smartphones and landlines. The speech translation is performed incrementally based on generation of partial hypotheses from speech recognition. Our system displays the recognized and translated text in an incremental fashion. The use of SIP-based technology also supports an open form of cross-lingual dialog without the need for attention phrases.

3 Incremental Speech-to-Speech Translation

In most statistical machine translation systems, the input source text is translated in entirety, i.e., the search for the optimal target string is constrained on the knowledge of the entire source string. However, in applications such as language learning and real-time speech-to-speech translation, incrementally translating the source text or speech can provide seamless communication and understanding with low latency. Let us assume that the input string (either text or speech recognition hypothesis) is $\mathbf{f} = f_1, \cdots, f_J$ and the target string is $\mathbf{e} = e_1, \cdots, e_I$. Among all possible target sentences, we will choose the one with highest probability:

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

In an incremental translation framework, we do not observe the entire string $\mathbf{f}$. Instead, we observe Q_s sequences, $\mathbf{S} = s_1 \cdots s_k \cdots s_{Q_s}$, i.e., each sequence $s_k = [f_{j_k} f_{j_k+1} \cdots f_{j_{(k+1)}-1}]$, $j_1 = 1, j_{Q_s+1} = J + 1$[1]. Let the translation of each foreign sequence s_k be denoted by $t_k = [e_{i_k} e_{i_k+1} \cdots e_{i_{(k+1)}-1}]$, $i_1 = 1, i_{Q_s+1} = I+1$. Given this setting, we can perform decoding using three different approaches. Assuming that each partial source input is translated independently, i.e., chunk-wise translation, we get,

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

We call the decoding in Eq. 2 as *partial* decoding. The other option is to translate the partial source in-

[1] For simplicity, we assume that the incremental and non-incremental hypotheses are equal in length

put conditioned on the history, i.e.,

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

where t_i^* denotes the best translation for source sequence s_i. We term the result obtained through Eq. 3 as *continue-partial*. The third option is to wait for all the partials to be generated and then decode the source string which we call *complete* decoding, i.e.,

$$\hat{\hat{\mathbf{e}}}(\mathbf{f}) = \arg\max_{\mathbf{e}} \Pr(\mathbf{e}|s_1, \cdots, s_k) \quad (4)$$

Typically, the hypothesis $\hat{e}$ will be more accurate than $\hat{\hat{e}}$ as the translation process is non-incremental. In the best case, one can obtain $\hat{\hat{e}} = \hat{e}$. While the decoding described in Eq. 2 has the lowest latency, it is likely to result in inferior performance in comparison to Eq. 1 that will have higher latency. One of the main issues in incremental speech-to-speech translation is that the translated sequences need to be immediately synthesized. Hence, there is tradeoff between the amount of latency versus accuracy as the synthesized audio cannot be revoked in case of long distance reordering. In this work, we focus on incremental speech translation and defer the problem of incremental synthesis to future work. We investigate the problem of incrementality using a novel SIP-based S2S translation system, the details of which we discuss in the subsequent sections.

4 Speech-to-Speech Translation Components

In this section, we describe the training data, preprocessing steps and statistical models used in the S2S system.

4.1 Automatic Speech Recognition

We use the AT&T WATSON$^{\mathrm{SM}}$ real-time speech recognizer (Goffin et al., 2004) as the speech recognition module. WATSON$^{\mathrm{SM}}$ uses context-dependent continuous density hidden Markov models (HMM) for acoustic modeling and finite-state networks for network optimization and search. The acoustic models are Gaussian mixture tied-state three-state left-to-right HMMs. All the acoustic models in this work were initially trained using the Maximum Likelihood Estimation (MLE) criterion, and followed by discriminative training through Minimum Phone Error (MPE) criterion. We also employed Gaussian Selection (Bocchieri, 1993) to decrease the real-time factor during the recognition procedure.

The acoustic models for English and Spanish were mainly trained on short utterances in the respective language, acquired from SMS and search applications on smartphones. The amount of training data for the English acoustic model is around 900 hours of speech, while the data for training the Spanish is approximately half that of the English model. We used a total of 107 phonemes for the English acoustic model, composed of digit-specific, alpha-specific, and general English phonemes. Digit-specific and alpha-specific phonemes were applied to improve the recognition accuracy of digits and alphas in the speech. The number of phonemes for Spanish was 34, and, no digit- or alpha-specific phonemes were included. The pronunciation dictionary for English is a hand-labeled dictionary, with pronunciation for unseen words being predicted using custom rules. A rule-based dictionary was used for Spanish.

We use AT&T FSM toolkit (Mohri et al., 1997) to train a trigram language model (LM). The language model was linearly interpolated from 18 and 17 components for English and Spanish, respectively. The data for the the LM components was obtained from several sources that included LDC, Web, and monolingual portion of the parallel data described in section 4.2. An elaborate set of language specific tokenization and normalization rules was used to clean the corpora. The normalization included spelling corrections, conversion of numerals into words while accounting for telephone numbers, ordinal, and, cardinal categories, punctuation, etc. The interpolation was performed by tuning the language model weights on a development set using perplexity metric. The development set was 500 sentences selected randomly from the IWSLT corpus (Paul, 2006). The training vocabulary size for English acoustic model is 140k and for the language model is 300k. For the Spanish model, the training vocabulary size is 92k, while for testing, the language model includes 370k distinct words. In our experiments, the decoding and LM vocabularies were the same.

4.2 Machine Translation

The phrase-based translation experiments reported in this work was performed using the Moses[2] toolkit (Koehn et al., 2007) for statistical machine translation. Training the translation model starts from the parallel sentences from which we learn word alignments by using GIZA++ toolkit (Och and Ney, 2003). The bidirectional word alignments obtained using GIZA++ were consolidated by using the *grow-diag-final* option in Moses. Subsequently, we learn phrases (maximum length of 7) from the consolidated word alignments. A lexicalized reordering model (*msd-bidirectional-fe* option in Moses) was used for reordering the phrases in addition to the standard distance based reordering (*distortion-limit* of 6). The language models were interpolated Kneser-Ney discounted trigram models, all constructed using the SRILM toolkit (Stolcke, 2002). Minimum error rate training (MERT) was performed on a development set to optimize the feature weights of the log-linear model used in translation. During decoding, the unknown words were preserved in the hypotheses.

The parallel corpus for phrase-based translation was obtained from a variety of sources: europarl (Koehn, 2005), jrc-acquis corpus (Steinberger et al., 2006), opensubtitle corpus (Tiedemann and Lars Nygaard, 2004), web crawling as well as human translation. The statistics of the data used for English-Spanish is shown in Table 1. About 30% of the training data was obtained from the Web (Rangarajan Sridhar et al., 2011). The development set (identical to the one used in ASR) was used in MERT training as well as perplexity based optimization of the interpolated language model. The language model for MT and ASR was constructed from identical data.

4.3 Text-to-speech synthesis

The translated sentence from the machine translation component is synthesized using the AT&T Natural Voices$^{\text{TM}}$ text-to-speech synthesis engine (Beutnagel et al., 1999). The system uses unit selection synthesis with half phones as the basic units. The database was recorded by professional speakers of the language. We are currently using female voices for English as well as Spanish.

	en-es	
Data statistics	en	es
# Sentences	7792118	7792118
# Words	98347681	111006109
Vocabulary	501450	516906

Table 1: Parallel data used for training translation models

5 SIP Communication Framework for Real-time S2S Translation

The SIP communication framework for real-time language translation comprises of three main components. Session Initiation Protocol (SIP) is becoming the de-facto standard for signaling control for streaming applications such as Voice over IP. We present a SIP communication framework that uses Real-time Transport Protocol (RTP) for packetizing multimedia content and User Datagram Protocol (UDP) for delivering the content. In this work, the content we focus on is speech and text information exchanged between two speakers in a cross-lingual dialog. For two users conversing in two different languages (e.g., English and Spanish), the media channels between them will be established as shown in Figure 1. In Figure 1, each client (UA) is responsible for recognition, translation, and synthesis of one language input. E.g., the English-Spanish UA recognizes English text, converts it into Spanish, and produces output Spanish audio. Similarly, the Spanish-English UA is responsible for recognition of Spanish speech input, converting it into English, and producing output English audio. We describe the underlying architecture of the system below.

5.1 Architecture

1. End point SIP user agents: These are the SIP end points that exchange SIP signaling messages with the SIP Application server (AS) for call control.

2. SIP User Agents: Provide a SIP interface to the core AT&T WATSON$^{\text{SM}}$ engine that incorporates acoustic and language models for speech

[2] http://www.statmt.org/moses

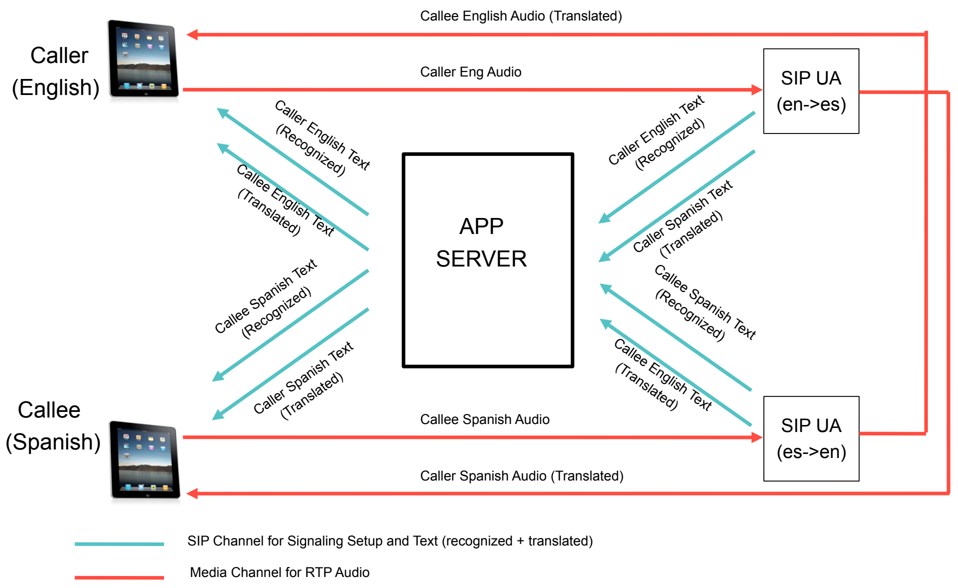


Figure 1: SIP communication framework used for real-time speech-to-speech translation. The example shows the setup between two participants in English(en) and Spanish (es)

recognition.

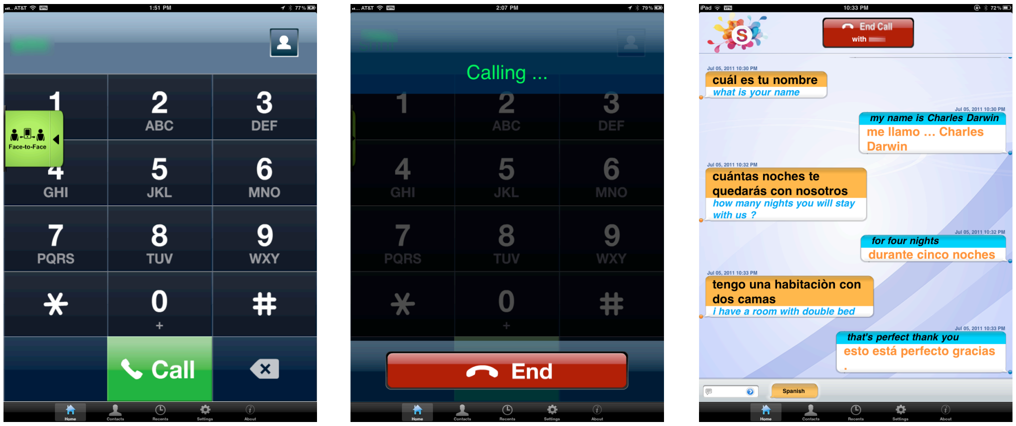
3. SIP Application Server (AS): A standard SIP B2BUA (back to back user agent) that receives SIP signaling messages and forwards them to the intended destination. The machine translation component (server running Moses (Koehn et al., 2007)) is invoked from the AS.

In our communication framework, the SIP AS receives a call request from the calling party. The AS infers the language preference of the calling party from the user profile database and forwards the call to the called party. Based on the response, AS infers the language preference of the called party from the user profile database. If the languages of the calling and called parties are different, the AS invites two SIP UAs into the call context. The AS exchanges media parameters derived from the calling and called party SIP messages with that of the SIP UAs. The AS then forwards the media parameters of the UAs to the end user SIP agents.

The AS, the end user SIP UAs, and the SIP UAs are all RFC 3261 SIP standard compliant. The end user SIP UAs are developed using PJSIP stack that uses PJMedia for RTP packetization of audio and network transmission. For our testing, we have implemented the end user SIP UAs to run on Apple IOS devices. The AS is developed using E4SS (Echarts for SIP Servlets) software and deployed on Sailfin Java container. It is deployed on a Linux box installed with Cent OS version 5. The SIP UAs are written in python for interfacing with external SIP devices, and use proprietary protocol for interfacing with the core AT&T WATSON^{SM} engine.

6 Typical Call Flow

Figure 2 shows the typical call flow involved in setting up the cross-lingual dialog. The caller chooses the number of the callee from the address book or enters it using the keypad. Subsequently, the call is initiated and the underlying SIP channels are established to facilitate the call. The users can then converse in their native language with the hypotheses displayed in an IM-like fashion. The messages of the caller appear on the left side of the screen while those of the callee appear on the right. Both the recognition and translation hypotheses are displayed incrementally for each side of the conversation. In our experiments, the caller and the callee naturally followed a protocol of listening to the other party's synthesized output before speaking once they were accustomed to the interface. One of the issues during speech recognition is that, the user can potentially start speaking as the TTS output from the other



Jul 05, 2011 10:30 PM
cuál es tu nombre
what is your name

Jul 05, 2011 10:30 PM
my name is Charles Darwin
me llamo ... Charles Darwin

Jul 05, 2011 10:32 PM
cuántas noches te quedarás con nosotros
how many nights you will stay with us ?

Jul 05, 2011 10:32 PM
for four nights
durante cinco noches

Jul 05, 2011 10:33 PM
tengo una habitaciòn con dos camas
i have a room with double bed

Jul 05, 2011 10:33 PM
that's perfect thank you
esto está perfecto gracias
.

Figure 2: Illustration of call flow. The call is established using SIP and the real-time conversation appears in the bubbles in a manner similar to Instant Messaging. For illustration purposes, the caller (Spanish) and callee (English) are assumed to have set their language preferences in the setup menu.

participant is being played. We address the feedback problem from the TTS output by muting the microphone when TTS output is played.

7 Dialog Data

The system described above provides a natural way to collect cross-lingual dialog data. We used our system to collect a corpus of 40 scripted dialogs in English and Spanish. A bilingual (English-Spanish) speaker created dialog scenarios in the travel and hospitality domain and the scripted dialog was used as reference material in the call. Two subjects participated in the data collection, a male English speaker and female Spanish speaker. The subjects were instructed to read the lines verbatim. However, due to ASR errors, the subjects had to repeat or improvise few turns (about 10%) to sustain the dialog. The average number of turns per scenario in the collected corpus is 13; 6 and 7 turns per scenario for English and Spanish, respectively. An example dialog between two speakers is shown in Table 2.

A: Hello, I am calling from room four twenty one the T.V. is not working. Do you think you can send someone to fix it please?
B: Si, Señor enseguida enviamos a alguien para que la arregle. Si no le cambiaremos de habitación.
A: Thank you very much.
B: Estamos aqu para servirle. Llámenos si necesita algo más.

Table 2: Example of a sample dialog scenario.

8 Experiments

In this section, we describe speech translation experiments performed on the dialog corpus collected through our system. We present baseline results followed by results of incremental translation.

8.1 Baseline Experiments

The models described in Section 4 were used to establish baseline results on the dialog corpus. No contextual information was used in these experiments, i.e., the audio utterances were decoded independently. The ASR WER for English and Spanish sides of the dialogs is shown in Figure 3. The average WER for English and Spanish side of the conversations is 27.73% and 22.83%, respectively. The recognized utterances were subsequently translated using the MT system described above. The MT performance in terms of Translation Edit Rate (TER) (Snover et al., 2006) and BLEU (Papineni et al., 2002) is shown in Figure 4. The MT performance is shown across all the turns for both reference transcriptions and ASR output. The results show that the performance of the Spanish-English MT model is better in comparison to the English-Spanish model on the dialog corpus. The performance on ASR input drops by about 18% compared to translation on reference text.

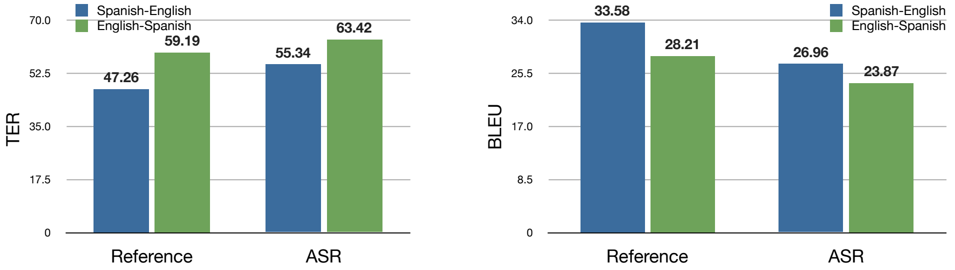


Figure 4: TER (%) and BLEU of English-Spanish and Spanish-English MT models on reference transcripts and ASR output

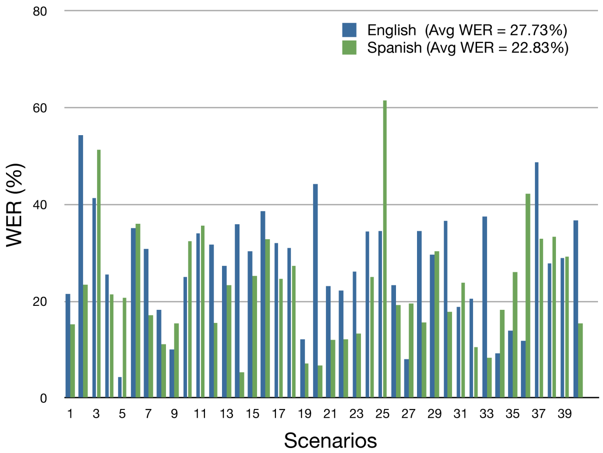


Figure 3: WER (%) of English and Spanish acoustic models on the dialog corpus

8.2 Segmentation of ASR output for MT

Turn taking in a dialog typically involves the subjects speaking one or more utterances in a turn. Since, machine translation systems are trained on chunked parallel texts (40 words or less), it is beneficial to segment the ASR hypotheses before translation. Previous studies have shown significant improvements in translation performance through the segmentation of ASR hypotheses (Matusov et al., 2007). We experimented with the notion of segmentation defined by silence frames in the ASR output. A threshold of 8-10 frames (100 ms) was found to be suitable for segmenting the ASR output into sentence chunks. We did not use any lexical features for segmenting the turns. The BLEU scores for different silence thresholds used in segmentation is shown in Figure 5. The BLEU scores improvement for Spanish-English is 1.6 BLEU points higher than the baseline model using no segmentation. The improvement for English-Spanish is smaller but statistically significant. Analysis of the dialogs revealed that the English speaker tended to speak his turns without pausing across utterance chunks while the Spanish speaker paused a lot more. The results indicate that in a typical dialog interaction, if the participants observe inter-utterance pause (80-100 ms) within a turn, it serves as a good marker for segmentation. Further, exploiting such information can potentially result in improvements in MT performance as the model is typically trained on sentence level parallel text.

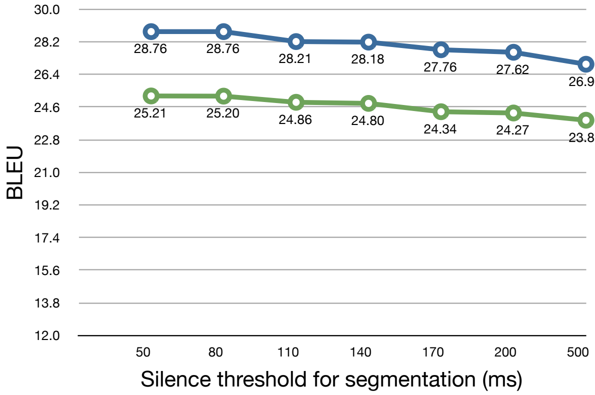


Figure 5: BLEU score of English-Spanish and Spanish-English MT models on the ASR output using silence segmentation

8.3 Incremental Speech Translation Results

Figure 6 shows the BLEU score for incremental speech translation described in Section 3. In the figure, *partial* refers to Eq. 2, *continue-partial* refers to Eq. 3 and *complete* refers to Eq. 4. The continue-partials option was exercised by using the continue-

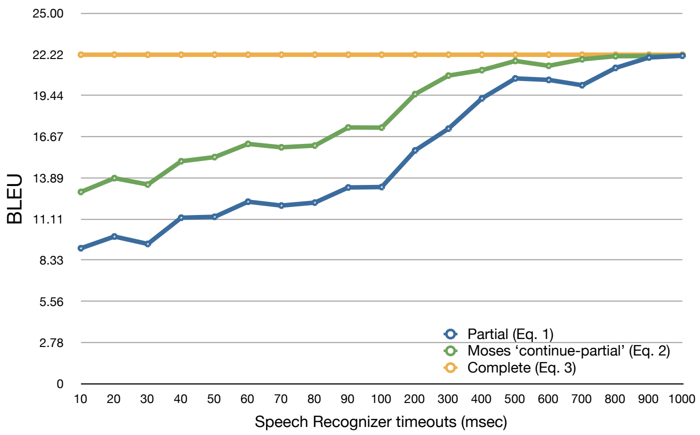


Figure 6: BLEU score (Spanish-English) for incremental speech translation across varying timeout periods in the speech recognizer

partial-translation parameter in Moses (Koehn et al., 2007). The partial hypotheses are generated as a function of speech recognizer timeouts. Timeout is defined as the time interval with which the speech recognizer generates partial hypotheses. For each timeout interval, the speech recognizer may or may not generate a partial result based on the search path at that instant in time. As the timeout interval increases, the performance of incremental translation approaches that of non-incremental translation. The key is to choose an operating point such that the user perception of latency is minimal with acceptable BLEU score. It is interesting that very good performance can be attained at a timeout of 500 ms in comparison with non-incremental speech translation, i.e., the latency can be reduced in half with acceptable translation quality. The *continue-partial* option in Moses performs slightly better than the *partial* case as it conditions the decision on prior source input as well as translation.

In Table 3, we present the latency measurements of the various components in our framework. We do not have a row for ASR since it is not possible to get the start time for each recognition run as the RTP packets are continuously flowing in the SIP framework. The latency between various system components is very low (5-30 ms). While the average time taken for translation (incremental) is ≈ 100 ms, the TTS takes the longest time as it is non-incremental in the current work. It can also been seen that the average time taken for generating incremental MT output is half that of TTS that is non-incremental. The overall results show that the communication in our SIP-based framework has low latency.

Components	Caller	Callee	Average
ASR output to MT input	6.8	0.1	3.4
MT	100.4	108.8	104.6
MT output to TTS	22.1	33.1	27.6
TTS	246	160.3	203.1

Table 3: Latency measurements (in ms) for the S2S components in the real-time SIP framework.

9 Conclusion

In this paper, we introduced the problem of incremental speech-to-speech translation and presented first of a kind two-way real-time speech-to-speech translation system based on SIP that incorporates the notion of incrementality. We presented details about the SIP framework and demonstrated the typical call flow in our application. We also presented a dialog corpus collected using our framework and benchmarked the performance of the system. Our framework allows for incremental speech translation and can provide low latency translation. We are currently working on improving the accuracy of incremental translation. We are also exploring new algorithms for performing reordering aware incremental speech-to-speech translation, i.e., translating source phrases such that text-to-speech synthesis can be rendered incrementally.

References

M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal. 1999. The AT&T Next-Gen TTS system. In *Proceedings of Joint Meeting of ASA, EAA and DEGA*.

E. Bocchieri. 1993. Vector quantization for the efficient computation of continuous density likelihoods. *Proceedings of ICASSP*.

Charles L. Chen and T. V. Raman. 2008. Axsjax: a talking translation bot using google im: bringing web-2.0 applications to life. In *Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A)*.

O. Furuse and H. Iida. 1996. Incremental translation utilizing constituent boundary patterns. In *Proc. of Coling '96*.

Vincent Goffin, Cyril Allauzen, Enrico Bocchieri, Dilek Hakkani Tur, Andrej Ljolje, and Sarangarajan Parthasarathy. 2004. The AT&T Watson Speech Recognizer. Technical report, September.

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*.

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

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

F. Metze, J. McDonough, H. Soltau, A. Waibel, A. Lavie, S. Burger, C. Langley, L. Levin, T. Schultz, F. Pianesi, R. Cattoni, G. Lazzari, N. Mana, and E. Pianta. 2002. The NESPOLE! speech-to-speech translation system.

M. Mohri, F. Pereira, and M. Riley. 1997. Att 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.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*.

M. Paul. 2006. Overview of the iwslt 2006 evaluation campaign. In *Proceedings of the International Workshop of Spoken Language Translation*, Kyoto, Japan.

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*.

B. Sankaran, A. Grewal, and A. Sarkar. 2010. Incremental decoding for phrase-based statistical machine translation. In *Proceedings of the fifth Workshop on Statistical Machine Translation and Metrics*.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA*.

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*.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of ICSLP*.

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

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

A. Waibel, A. N. Jain, A. E. McNair, H. Saito, A. G. Hauptmann, and J. Tebelskis. 1991. JANUS: a speech-to-speech translation system using connectionist and symbolic processing strategies. In *Proceedings of ICASSP*, pages 793–796, Los Alamitos, CA, USA.

A. Waibel, A. Badran, A. W. Black, R. Frederking, G. Gates, A. Lavie, L. Levin, K. Lenzo, L. M. Tomokiyo, J. Reichert, T. Schultz, W. Dorcas, M. Woszczyna, and J. Zhang. 2003. Speechalator: two-way speech-to-speech translation on a consumer PDA. In *Proceedings of the European Conference on Speech Communication and Technology*, pages 369–372.

B. Zhou, Y. Gao, J. Sorenson, D. Dechelotte, and M. Picheny. 2003. A hand-held speech-to-speech translation system. In *Proceedings of ASRU*.