

# INTEGRATING SPEECH RECOGNITION AND MACHINE TRANSLATION

Spyros Matsoukas, Ivan Bulyko, Bing Xiang, Kham Nguyen<sup>†</sup>, Richard Schwartz, John Makhoul

BBN Technologies  
10 Moulton St., Cambridge, MA 02138  
smatsouk@bbn.com

## ABSTRACT

This paper presents a set of experiments that we conducted in order to optimize the performance of an Arabic/English machine translation system on broadcast news and conversational speech data. Proper integration of speech-to-text (STT) and machine translation (MT) requires special attention to issues such as sentence boundary detection, punctuation, STT accuracy, tokenization, conversion of spoken numbers and dates to written form, optimization of MT decoding weights, and scoring. We discuss these issues, and show that a carefully tuned STT/MT integration can lead to significant translation accuracy improvements compared to simply feeding the regular STT output to a text MT system.

**Index Terms**— Speech Recognition, Machine Translation, Sentence Boundary Detection

## 1. INTRODUCTION

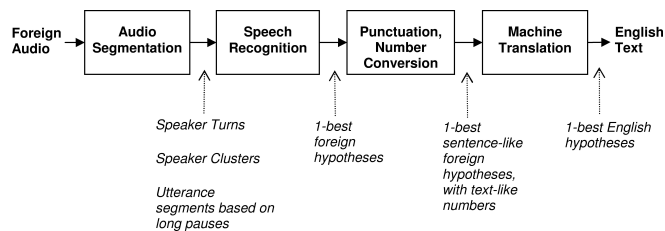
Most MT systems are trained on a bilingual corpus derived from text sources. In such a corpus, each foreign segment corresponds to a sentence, with properly placed punctuation marks. In addition, all occurrences of numbers, dates, monetary amounts, abbreviations, etc., in a foreign segment appear as in ordinary text. However, in an integrated STT/MT system, the MT component needs to process foreign STT output, which differs from text in several ways:

1. Each STT segment is generated by an automatic audio segmentation process which is based solely on acoustic cues (long pauses, speaker/channel variations). Therefore, the STT segments may not correspond to sentences in the foreign language.
2. Current STT output consists only of the spoken words, with no punctuation.
3. Numbers, dates, etc., appear in spoken form, e.g. “two hundred twenty five” instead of “225”.

Due to the above differences, integration of STT and MT is non-trivial, as the STT output needs to be processed prior to translation to (a) detect sentence boundaries, (b) add punctuation, and (c) convert foreign spoken numbers to digits. In addition, one needs to make sure that the STT output is tokenized and normalized in a manner that is consistent with the MT component. Such post-processing of the STT output is easiest when the integration is based on the 1-best STT output, rather than a speech lattice. Thus, our first experiments used 1-best STT output, resulting in the integration pipeline shown in Figure 1.

<sup>†</sup> Kham Nguyen is a Ph.D. student at the College of Computer & Information Science, Northeastern University, Boston, MA.

This work was supported under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022.



**Fig. 1.** The STT/MT integration pipeline investigated in this set of experiments

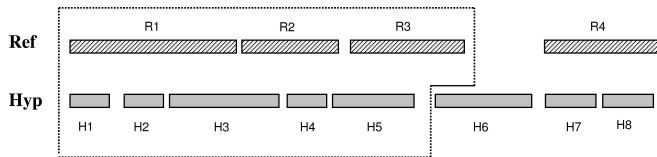
The rest of the paper is organized as follows: Section 2 describes the procedure used to evaluate MT performance in the experiments presented here. In Section 3, we describe the experimental setup and show the effect of STT segmentation, accuracy, and punctuation on MT performance. Section 4 discusses methods for improved sentence boundary detection. The problem of optimizing MT decoding parameters on speech data is addressed in Section 5. The paper ends with some conclusions and future work discussion in Section 6.

## 2. SCORING MT OUTPUT

It is worth mentioning that the work presented in this paper was performed for the Global Autonomous Language Exploitation (GALE) program, sponsored by the Defense Advanced Research Project Agency (DARPA). Typically, MT systems are evaluated in terms of BLEU [1], however, recent studies have shown that Translation Edit Rate (TER) [2] correlates better with human-mediated TER (HTER), the primary MT performance evaluation metric used under GALE. Thus, all the results reported in this paper are based on TER scoring.

Applying TER scoring to the output of STT/MT integration is not trivial, since the hypothesis and reference translations have different segmentations when translating from automatically segmented speech, and as a result there is no one-to-one correspondence between hypothesis and reference segments. In this case, in order to compute TER properly, one has to group adjacent hypothesis and reference segments into “chunks” based on their start/end times, such that both hypothesis and reference agree on the boundaries within a chunk. This illustrated in the example of Figure 2, where hypothesis segments H1-H5 were grouped together with reference segments R1-R3, forming the chunk enclosed by the dotted line.

Notice that in this example, hypothesis segment H6 overlaps mostly with a gap in the reference segmentation, so H6 is excluded from TER scoring. We have implemented a set of tools that automate this scoring procedure. The algorithm for the chunk construction is given below:



**Fig. 2.** Aligning hypothesis and reference segments into chunks, in order to perform TER calculation

1. For each hypothesis segment, compute its overlap with all reference segments as a percentage of its duration. If the overlap is less than a threshold (e.g., 50%), then exclude this hypothesis segment from scoring. This step is needed in order to avoid insertion errors in regions that are excluded from scoring (e.g., commercials and other non-news segments are typically excluded when scoring translation of broadcast news speech).
2. For each reference segment, find all non-excluded hypothesis segments that it overlaps with.
3. Loop over the reference segments, and for each pair of adjacent segments compute the intersection of their corresponding overlapping hypothesis sets. If the intersection is non-empty, there is a hypothesis segment that crosses the boundary between the two reference segments (e.g., segment H3 in Figure 2). In this case we need to include both reference segments and their related hypotheses as part of the same chunk for TER scoring. If, on the other hand, the intersection is empty, we can insert a chunk boundary between the two reference segments (e.g., between segments R3 and R4 in Figure 2). This process is repeated until all reference segments have been examined.

Once the chunks have been defined, hypothesis segments within each chunk are merged together and are aligned against their corresponding concatenated reference segments, for the purpose of computing the chunk-level TER score. These scores are then averaged properly across the test set to compute the overall TER. This is the process that was used to score all the results reported in the tables that follow.

### 3. EFFECT OF STT SEGMENTATION, ACCURACY AND PUNCTUATION ON MT

As mentioned previously, our first integration experiments ran translation on the 1-best STT output. All experiments were performed on bnat05, a 6-hour Arabic Broadcast News (BN) development test set compiled from several sources from January 2001 and November 2003. Scoring of the STT output was done using NIST tools, after normalizing hypothesis and reference files using the same normalization script used to process the Arabic side of the MT training bitext. Scoring of the MT output was based on the TER procedure described previously, with mixed-case text.

To investigate the effect of STT accuracy on MT performance, we made use of three STT systems, described below:

**STT-A:** BBN EARS RT04 Arabic BN grapheme system [3], with acoustic model trained on 100 hours of speech and a trigram language model trained on 400 million words of news text.

**STT-B:** Uses morphological analyzer and automatic methods to infer short vowels in word pronunciations. Makes use of additional acoustic training data (50 hours).

**STT-C:** Like STT-B, but with additional language model training data.

We also used two versions of the BBN MT system, listed below.

**MT-A:** System developed at BBN during the period Sep 2004 - Apr 2005. Phrase-based statistical MT model, trained on 100M words of Arabic/English UN and news bitext. Trigram English LM, trained on 2 billion words of text (mostly newswire). Translation based on posterior probability of English given Foreign.

**MT-B:** Uses a combination of generative and posterior translation probabilities, includes a phrase segmentation score, and uses a method to compensate for over-estimated translation probabilities. Optimizes decoding weights by minimizing TER on N-best lists, using a process similar to [4].

The TER scores of systems MT-A and MT-B on the 2002 NIST MT evaluation test set were 48.29% and 46.35%, respectively.

Table 1 shows integration results obtained on the bnat05 test set, using various combinations of STT and MT systems. This table also shows the effect of segmentation and punctuation of STT output on MT performance.

STT System	MT System	STT WER	Segm.	Punc.	TER
STT-A	MT-A	22.2	automatic	period	66.76
STT-B	MT-A	18.3	automatic	period	65.95
STT-B	MT-B	18.3	automatic	period	64.55
STT-B	MT-B	17.6	reference	period	61.89
N/A	MT-B	0.0	reference	period	58.67
N/A	MT-B	0.0	reference	reference	57.97

**Table 1.** Results on the bnat05 test set, showing the effect of speech recognition accuracy, segmentation, and punctuation on TER

We can see that even though system STT-B is significantly better than STT-A in terms of WER, the improvement in TER after translating with system MT-A is less than 1% absolute. Improving the MT component, on the other hand, has a larger effect on the TER score. In the fourth-row experiment we used the reference segmentation for both STT and MT. Interestingly, this resulted in a small WER improvement, but a larger TER reduction. If instead of the STT output we use the true reference Arabic transcript (0.0% WER), there is approximately 3% absolute gain in TER, which indicates that at the current level of MT performance, dramatic improvements in STT accuracy have only modest effect on TER. All these experiments used minimal punctuation in the MT input (only an ending period in each segment). In the last experiment, the full reference punctuation was used (commas, quotation marks, etc.), however this resulted in only a small TER improvement, suggesting that accurate punctuation of the source is not important at this level of MT performance.

In order to better understand the effect of STT segmentation on TER, we varied the parameters of our automatic audio segmentation procedure, as shown in Table 2. The segmentation procedure has two main parameters, the minimum duration of silence to consider for chopping the audio, and the maximum duration of a chopped segment.

We can see that "auto-seg-4" has characteristics similar to the reference segmentation, "ref-seg". All automatic segmentation results in Table 1 were obtained using segmentation "auto-seg-1". A comparison of the various automatic segmentations in terms of TER

Seg.	Min. Sil. Dur. (sec)	Max. Seg. Length (sec)	#Segments	Avg. Seg. Length (sec)
auto-seg-1	0.15	9	3427	6.17
auto-seg-2	0.30	15	2226	9.47
auto-seg-3	0.30	20	1813	11.61
auto-seg-4	0.34	25	1548	13.60
ref-seg	N/A	N/A	1462	13.58

**Table 2.** Adjusting the parameters of the BBN automatic audio segmentation procedure on the bnat05 test set in order to increase the average segment length, trying to match the characteristics of the reference segmentation (ref-seg)

is shown in Table 3. This comparison was made using system STT-C for generating the STT output. The results show that the best TER score is obtained with the longer segmentation, gaining about 1.6% absolute compared to "auto-seg-1". Recall from Table 1 that the TER gain from reference segmentation was about 2.6% absolute (64.55 to 61.89). Thus, by adjusting the automatic segmentation we were able to obtain most of that gain.

STT System	STT WER	Segmentation	TER
STT-B	18.3	auto-seg-1	64.55
STT-C	17.8	auto-seg-1	64.40
STT-C	17.7	auto-seg-2	63.05
STT-C	17.7	auto-seg-3	62.89
STT-C	17.7	auto-seg-4	62.80

**Table 3.** Results on the bnat05 test set, showing the effect of various automatic audio segmentations on TER. System MT-B was used for translation in all experiments

The reason for the large effect of segmentation on TER is due to the fact that when a sentence boundary is misplaced in the hypothesis, the system is penalized with several errors, due to the mixed-case scoring. Additional errors may incur due to pure translation errors, since the MT system has been trained and expects to operate on sentence-like segments. The issue of optimal STT segmentation for the purposes of MT seems to be very important, and therefore we have explored an alternate segmentation approach, which is guided by both acoustic and linguistic cues and is described in Section 4.

It was previously mentioned that the STT output has numbers, dates, etc. in spoken form, while the MT system expects them in text form. All results reported in the above tables were obtained without any special processing of numbers in the STT output. Ideally, translation of numbers should be carried away using specialized components that detect the type of number (monetary amount, date, time, percentage, etc.) in the source language and convert it to the target language using the most appropriate formatting, given the surrounding context. One small step in that direction is to parse the STT output and convert all spoken numbers to digits. The effect of this STT post-processing step on TER is shown in Table 4.

#### 4. SENTENCE BOUNDARY DETECTION

The results of the previous section show that tuning acoustic segmentation to match the characteristics of reference segmentation helps, but this strategy may not generalize well on new test sets. For example, Broadcast Conversations (BC) exhibit more frequent speaker

Number Conversion	TER
No	62.80
Yes	62.37

**Table 4.** Results on bnat05, showing the effect of converting Arabic spoken numbers to digits prior to translation. Original STT output was generated by system STT-C, using segmentation auto-seg-4. MT-B was used for both translation experiments

changes and shorter sentence lengths on average. Therefore, it is preferable to use a sentence boundary detection (SBD) approach. A standard technique for SBD is based on a Hidden Event Language Model (HELM) [5], which tries to detect certain types of punctuation based on linguistic context.

Several sentence boundary detection systems were compared in terms of punctuation and translation performance. Table 5 provides brief description of the systems used. The baseline (system 1) simply inserts periods at the end of each acoustic segment. Systems 2, 3 and 4 discard the acoustic segmentation (with the exception of speaker turn boundaries where periods are forced), and make use of a 4-gram HELM trained on 769M words of Arabic news text. System 2 uses the HELM to insert periods within speaker turns. In system 3, the language model and silence durations are used jointly, by integrating silence duration as an observation into the HMM search. Finally, system 4 uses both language model and silence duration to insert periods at a high rate (about 50% higher than system 3) and then remove periods with the lowest model posteriors, while constraining the maximum sentence length (40 words). This approach helps overcome the "short-term memory" limitation of the HMM decoding search. Typically, HMM search considers n-gram contexts of not more than a few previous words (3 words in case of a 4-gram LM) making it hard to place restrictions on maximum sentence length.

System	Description
SBD-1	Acoustic segmentation baseline
SBD-2	Use only LM to insert periods within speaker turns
SBD-3	Use LM and silence jointly
SBD-4	Insert boundaries at a high rate using LM and silence, then remove the lowest scoring boundaries, constraining the maximum sentence length

**Table 5.** Descriptions of systems used for sentence boundary detection

Table 6 shows SBD and translation performance of various systems with respect to the baseline that uses acoustic segments with the average length of 9.47 seconds. Sentence boundaries produced with a language model alone (SBD-2) do not lead to improved performance. Adding the silence feature (SBD-3) improves the punctuation accuracy significantly, however this corresponds to only small improvements in translation. The best results are obtained by system SBD-4. Translation improvements on Modern Standard Arabic (MSA) regions are bigger than overall.

The experiments were repeated using a different set of recognition hypotheses where the average length of acoustic segments was 13.6 seconds (Table 7). In this case the SBD approach provided only a small additional gain in TER compared to the performance of auto-seg-4 segmentation. Nevertheless, the HELM-based SBD seems to be a safe operation on top of any type of acoustic segmentation.

System	SBD				TER (TER-MSA)
	Corr.	Del.	Ins.	Error	
SBD-1	66.71	33.29	73.03	106.30	62.55 (60.32)
SBD-2	59.44	40.56	57.86	98.42	62.66 (60.25)
SBD-3	68.98	31.02	60.40	91.42	62.49 (60.20)
SBD-4	63.97	36.03	50.10	86.14	62.32 (59.78)

**Table 6.** Segmentation and translation performance using STT output from system STT-C, with auto-seg-2 acoustic segmentation and spoken number conversion

System	SBD				TER (TER-MSA)
	Corr.	Del.	Ins.	Error	
SBD-1	56.97	43.03	38.64	81.67	62.37 (60.28)
SBD-2	58.27	41.73	58.13	99.86	62.81 (60.42)
SBD-3	67.40	32.60	57.17	89.77	62.79 (60.28)
SBD-4	70.49	29.51	61.43	90.94	62.34 (60.02)

**Table 7.** Segmentation and translation performance using STT output from system STT-C, with auto-seg-4 acoustic segmentation and spoken number conversion

## 5. OPTIMIZING MT ON SPEECH DATA

Recall that the MT component ranks hypotheses in the translation search space based on several knowledge sources, whose scores are combined log-linearly using a set of weights. In all integration experiments reported in the previous sections, the MT decoding weights were optimized to minimize TER on the NIST MT-02 set, consisting of newswire text. Broadcast speech has different style than newswire text, so it is reasonable to expect translation accuracy improvements from tuning the MT decoding weights on BN-type material.

Ideally, in a joint STT/MT system the MT decoding weights should be optimized to minimize TER based on translations of STT output. However, this gets complicated due to the scoring issues described in Section 2, so we considered the following procedure instead:

1. Run translation on the Arabic reference transcripts (punctuation included), after automatically converting spoken numbers to digits.
2. Tune weights on N-best from step 1, based on the reference translations. This typically requires a couple of iterations of step 1 and 2, until the weights converge.
3. Take the best weights from step 2 and use them to translate automatically segmented STT output.

This procedure is preferable because it eliminates the need to segment speech according to the reference segmentation and then run speech recognition on the resulting segments, to generate the 1-best output for translation. When using translations of reference transcripts for MT tuning, it is important to make sure that each segment in the STT reference transcripts corresponds to a single sentence. This is necessary because multi-sentence segments tend to be quite long, resulting in N-best translations where the 1-best hypothesis differs from the rest of the hypotheses in the N-best only in minor ways (one to two tokens). Such degenerate N-best is not a good representation of the full hypothesis search space, and could lead to sub-optimal MT tuning.

In all experiments described below, the decoding weights were tuned on the bnat06 (BN) and bcat06 (BC) sets, and then were tested

on bnat06 (BN) and bcat06 (BC). These sets total about 6 hours of speech broadcasted in January of 2006. We used STT output from models trained on 1000 hours Arabic audio data. The WERs of those experiments are listed in Table 8.

Test set	Usage	WER
bnat06	Tuning	21.0
bcat06		29.2
bnad06	Validation	20.1
bcad06		29.7

**Table 8.** STT output WERs on the Arabic speech tuning sets, bnat06 and bcat06, and on the validation sets bnad06 and bcad06

All results in Table 9 were obtained by translating BBN’s STT output on bnat06 and bcad06, with automatic segmentation. In each case, the MT system weights were tuned on the designated set (under column “OptSet”), with 3-4 iterations of optimization. Three optimization sets are compared: MT-02 set, translations of bnat06 and bcat06 reference transcripts (bnc-ref), and manually segmented (according to reference) bnat06 and bcat06 STT output (bnc-stt).

We can see that, although there is a clear benefit for tuning MT system weights on bnat06 and bcat06 (as opposed to using weights estimated on MT02), there is no significant difference between tuning on translations of STT output or reference transcripts.

OptSet	bnad06		bcad06	
	TER	BLEU	TER	BLEU
MT02	67.42	13.14	73.32	10.33
bnc-ref	66.78	15.07	71.85	12.19
bnc-stt	66.86	14.94	71.93	12.14

**Table 9.** Results on bnad06 (BN) and bcad06 (BC), showing how translation of automatically segmented STT output is affected by the optimization set used to tune MT decoding weights

## 6. CONCLUSIONS AND FUTURE WORK

This paper has addressed several issues that arise in the integration of STT and MT components. Experimental results show that significant translation accuracy improvements can be obtained by paying special attention to sentence segmentation, conversion of spoken numbers to written form, and MT decoding weight optimization. Future work will include translation of speech lattice output and joint optimization of STT and MT components.

## 7. REFERENCES

- [1] K. Papineni, *et al.*, “BLEU: a Method for Automatic Evaluation of Machine Translation,” in *ACL*, July 2002, pp. 311–318.
- [2] M. Snover, *et al.*, “A Study of Translation Edit Rate with Targeted Human Annotation,” in *AMTA*, 2006.
- [3] M. Afify, *et al.*, “Recent progress in Arabic Broadcast News transcription at BBN,” in *Interspeech*, Sept. 2005.
- [4] F. J. Och, “Minimum Error Rate Training in Statistical Machine Translation,” in *ACL*, July 2003, pp. 160–167.
- [5] A. Stolcke *et al.*, “Automatic detection of sentence boundaries and disfluencies based on recognized words,” in *ICSLP*, vol. 2, 1998, pp. 2247–2250.