THE RWTH ARABIC-TO-ENGLISH SPOKEN LANGUAGE TRANSLATION SYSTEM

Oliver Bender, Evgeny Matusov, Stefan Hahn, Saša Hasan, Shahram Khadivi, and Hermann Ney

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6 - Computer Science Department
RWTH Aachen University, D-52056 Aachen, Germany
{bender,matusov,hahn,hasan,khadivi,ney}@cs.rwth-aachen.de

ABSTRACT

We present the RWTH phrase-based statistical machine translation system designed for the translation of Arabic speech into English text. This system was used in the Global Autonomous Language Exploitation (GALE) Go/No-Go Translation Evaluation 2007.

Using a two-pass approach, we first generate n-best translation candidates and then rerank these candidates using additional models. We give a short review of the decoder as well as of the models used in both passes.

We stress the difficulties of spoken language translation, i.e. how to combine the recognition and translation systems and how to compensate for missing punctuation. In addition, we cover our work on domain adaptation for the applied language models. We present translation results for the official GALE 2006 evaluation set and the GALE 2007 development set.

Index Terms— speech to text, adjustment of ASR and MT vocabularies, LM adaptation, punctuation prediction

1. INTRODUCTION

We describe the RWTH spoken language translation system that was used in the Global Autonomous Language Exploitation (GALE)[1] Go/No-Go Translation Evaluation this summer. The system performs a two-pass approach: in the first pass our statistical phrase-based decoder generates n-best translation candidates which are reranked applying additional models in the second pass.

When going from text translation to the translation of automatically recognized speech, one has to focus on a couple of problems. First of all, it has to be ensured that both the automatic speech recognition (ASR) system and the statistical machine translation (SMT) system use matching vocabularies. Furthermore, ASR output lacks the existence of any type of punctuation marks or sentence segmentation. Case information is not present, numbers and abbreviations are written out as words, recognition errors occur, and one has to deal with the effects of natural speech like hesitations and filler words.

Within the GALE evaluations, the systems have to translate recorded speech out of two different domains: broadcast news (BN) and broadcast conversations (BC) which are focused more on discussions and call-ins that have a conversational style of speech. The most straightforward way to tailor an SMT system to a specific domain is to apply domain adapted language models.

The paper is organized as follows. Section 2 gives a brief overview of our SMT system and the models used in the two passes. Section 3 then reports on our work to adjust the ASR and SMT vocabularies. Afterwards, we describe the domain adaptation for the used language models and the prediction of punctuation marks in the translation process in Sections 4 and 5. Experiments on the GALE 2006 evaluation set and on the GALE 2007 development set are discussed in section 6. Section 7 concludes.

2. THE RWTH SMT SYSTEM

In statistical machine translation, we are given a source language sentence $f_1^J = f_1 \dots f_j \dots f_J$, which is to be translated into a target language sentence $e_1^I = e_1 \dots e_i \dots e_I$.

Among all possible target language sentences, we choose the sentence with the highest probability:

$$\hat{e}_1^{\hat{I}} = \underset{I, e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I | f_1^J) \right\} \quad (1)$$

The posterior probability $Pr(e_1^I | f_1^J)$ is modeled directly using a log-linear combination of several models [1]:

$$Pr(e_1^I | f_1^J) = \frac{\exp\left(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)\right)}{\sum_{e'^{I'}_1} \exp\left(\sum_{m=1}^{M} \lambda_m h_m(e'^{I'}_1, f_1^J)\right)} \quad (2)$$

The denominator is a normalization factor that depends only on the source sentence f_1^J. Therefore, we can omit it during

[1] http://www.arpa.mil/ipto/programs/gale/index.htm

the search process. As a decision rule, we obtain:

$$\hat{e}_1^{\hat{I}} = \underset{I, e_1^I}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \right\} \quad (3)$$

This approach is a generalization of the source-channel approach [2]. The model scaling factors λ_1^M are optimized w.r.t. the final translation quality measured by an error criterion [3].

The overall system is similar to the ones successfully used in recent evaluations [4, 5, 6]. For a more detailed system description the reader is referred to these publications.

2.1. Models used during decoding

In the first pass, we run our phrase-based decoder to generate the n-best translation candidates using a log-linear combination of the following models:

- phrase-based model:
  During decoding, the hypotheses are generated by concatenating target language phrases. The pairs of source and target phrases that are consistent with the word alignment are extracted from the bilingual training corpus as described in [7]. We then use relative frequencies to estimate the phrase translation probabilities. To obtain a more symmetric model, the phrase-based model is used for both translation directions.

- phrase count features:
  As rare phrases tend to be overestimated and errors can originate from erroneous translations in the training data and misaligned words, we include features based on the actual count of the phrase pair. We check if this count is below a specific threshold. We use three phrase count features with manually chosen thresholds ranging from 1.0 to 3.0.

- word-based lexicon model:
  Longer phrases are rare and therefore tend to be overestimated. We use a word-based lexicon model to smooth the phrase translation probabilities. The computation is similar to IBM model 1 but only takes into account the words within the phrase pair. Like the phrase-based model, this model is used in both translation directions.

- word and phrase penalty model:
  These two models are simple heuristics which influence the average sentence and phrase lengths and can thus be used to enable the decoder to generate longer translation candidates.

- target language model:
  We apply a 4-gram language model which is built using the SRI Language Modeling toolkit [8] (smoothing technique is modified Kneser-Ney discounting with interpolation).

- reordering model:
  This model assigns costs simply based on the jump width, also used in, e.g. [9].

2.2. Rescoring models

Afterwards, we rerank the generated n-best translation candidates applying the following rescoring models:

- IBM model 1:
  This rescoring model measures the quality of the translations by using the IBM model 1 lexicon probabilities estimated during the word alignment training on a sentence level.

- deletion model:
  During IBM model 1 rescoring, we count all source words whose lexical probability given each target word is below a specified threshold, in the experiments the threshold was chosen between 10^{-1} and 10^{-4}.

- sentence length model:
  As described in [10], we explicitly model the target sentence length I by summing up the posterior probabilities of those target candidates that have length I.

- count language models:
  We apply on-the-fly language model estimation from n-gram counts using deleted interpolation. In the experiments, the Google n-gram counts and counts collected on the GigaWord corpus are used. We use 5-grams for this rescoring model.

3. ADJUSTMENT OF ASR AND SMT VOCABULARIES

As in any data-driven approach, our SMT system requires the proper preprocessing of training and testing data. Otherwise, the system will encounter many words that have not been observed in training and that are thus missing from the phrase tables and word lexica. When translating automatically recognized speech, preprocessing becomes even more important as the ASR and SMT systems are usually trained on different training data using different preprocessing tools. To overcome these difficulties and to adjust the ASR and SMT vocabularies, we perform the following steps. Note that both systems process UTF-8 encoded data.

1. First, we apply some rule-based normalization of Arabic words as described in [11], e.g. always mapping the hamza at the beginning of a word onto the same form, or removing the tanween character at the end of a word.

2. The next step is to split pre- and suffixes. For morphologically rich languages like Arabic, this step is important to reduce the number of occurring words and to obtain a computationally manageable system. In former

experiments [5], we used the MADA tool [12] for morphological disambiguation and applied the D2-scheme of [13] for word segmentation. These tools require the input to be Buckwalter encoded. Buckwalter maps the Arabic (UTF-8) characters onto an ASCII alphabet and is thus error prone since there may still be English words and non-Arabic characters in the original input which can not be represented in the Buckwalter encoding. We therefore decided to extract all splittings of pre- and suffixes on a Buckwalter encoded and MADA preprocessed version of the training data, to recode the splittings in UTF-8, and to apply them as mappings.

3. In a third step, the spoken numbers are converted to digits and regular expressions are used to categorize numbers, URLs and e-mail addresses.

4. DOMAIN ADAPTED LANGUAGE MODELS

In this section, we describe the training of language models resulting in genre-specific domain adaptation for the overall MT system. For the LM training, we combine different corpora representing various genres, e.g. broadcast news or conversations. The interpolation weights for these corpora are determined with the Downhill-Simplex algorithm, which is a standard approach for training the parameters of the log-linear model combination. The optimization criterion is the perplexity of the interpolated LM on a development set.

4.1. Implementation

We use the SRI Language Modeling toolkit [8] and incorporate the Downhill-Simplex method from the Numerical Recipes [14]. The genre-specific training corpora are separately loaded as dynamic language models where the interpolated probabilities of n-grams are calculated on-the-fly. This results in very efficient training of the interpolation weights, i.e. only the probabilities accessed during perplexity calculations are merged for the different LMs. Depending on the number of models, the training converges after 30–40 iterations.

In a final step, a static mixture of the LM is created and written to disk. Thus, it is possible to train several tuned baseline models (e.g. additional ones using more data) and again interpolate them using this approach. Interestingly, when applied to the specific genres such as BN and BC, the perplexity reductions on the development set carry over nicely to the test set, which makes this method appealing.

4.2. Results

In Table 1, the perplexity reductions on the test set are shown for the two domains. The baseline denotes perplexities obtained with a standard modified Kneser-Ney discounted language model without any genre-specific tuning (BASE) and trained on the whole target language corpora of the available bilingual data (GALE allowed corpora). After tuning the weights for each of the six main sub-parts of the LM and for each genre via Downhill-Simplex, we obtain significant reductions in terms of perplexity (DMIX-GS). As there is additional monolingual data available (e.g. GigaWord v2, TDT, BBN data, ...), this procedure is repeated, resulting in column DMIX-GS*. DMIX-GS* is tuned using DMIX-GS plus five additional (genre-specific) LMs.

	4-gram w/ KN discounting			
	BASE	DMIX-GS	DMIX-GS*	total red.
TEST-BC	116.3	95.9	90.5	-22.2%
TEST-BN	127.8	108.5	103.4	-19.1%

Table 1. Perplexities on the test set (GALE 2006 MT evaluation set) for various settings: BASE – baseline 4-gram w/ KN discounting; DMIX-GS – genre-specific adaptation using DS; DMIX-GS* – genre-specific adaptation using additional data.

We achieve overall reductions of 22% on BN and 19% on BN. The effect on the translation error measures can be seen in Section 6.

5. PUNCTUATION PREDICTION

The translations of the ASR output are expected to have proper sentence boundaries and punctuation. However, this annotation can not be transferred from the automatic transcripts, since the raw ASR output is just a sequence of words for a given audio document. We perform the sentence segmentation using the algorithm of ICSI/UW [15], which applies multiple acoustic and language model features to compute posterior probabilities of a segment boundary after each word. If the segmentation posterior probability is higher than a given threshold, a segment boundary is inserted. The threshold is optimized on a development set. Alternatively, we can use the dynamic programming algorithm of [16] with the posterior probability as the main feature. This algorithm has the advantage that the minimum and maximum sentence lengths can be explicitly specified, and an explicit language-specific sentence length model can be used. A limit on the maximum sentence length – 60 words – is necessary to reduce the computational complexity of translation. We set the minimum sentence length to 3 words, since one- and two-word segments are difficult to translate because of the missing context.

The ICSI/UW sentence segmentation system is able to predict the sentence type, i. e. if a sentence is a statement or a question. This information is used to generate the sentence-final punctuation – a period or a question mark. We insert these punctuation marks into the ASR output and translate them as usual.

In order to obtain sentence-internal punctuation in the English translations, we let the MT system predict the commas, as described in [16]. To this end, we train the word align-

		ARABIC	ENGLISH
TRAIN:	Sentences	7 M	
	Running Words	176 M	181 M
	Vocabulary	681 K	492 K
	Singletons	304 K	243 K
DEV-BC:	Sentences	315	
	Running Words	7 707	10 009
	Vocabulary	2 557	1 833
	OOVs	590	70
DEV-BN:	Sentences	565	
	Running Words	13 424	17 729
	Vocabulary	4 587	3 075
	OOVs	292	186
TEST-BC:	Sentences	529	
	Running Words	13 033	17 073
	Vocabulary	3 915	2 528
	OOVs	332	246
TEST-BN:	Sentences	956	
	Running Words	13 397	18 204
	Vocabulary	4 457	2 965
	OOVs	281	272

Table 2. Corpus Statistics of the GALE 2007 MT training data (TRAIN), the GALE 2007 MT development set (DEV-BC/BN), and the GALE 2006 MT evaluation set (TEST-BC/BN) after preprocessing.

ment as usual with punctuation marks present in the source and target part of the bilingual training corpus. Then, we remove all of the sentence-internal punctuation marks from the source part of the corpus only, adjusting the word alignment indices. Thus, many bilingual phrases extracted from the modified alignment contain target language commas (and other sentence-internal punctuation like semicolon) as insertions. During decoding, the decision on whether or not to insert a comma is made jointly by the translation model and the language model. The scaling factors of the MT models are re-optimized on the ASR output for the development set.

6. EXPERIMENTAL RESULTS

6.1. Experimental setup

Experiments are carried out on the current GALE MT test sets. In this work, we focus on the translation of Arabic transcripts out of the BC and BN domain. The corpus statistics are shown in Table 2. We train our models on approximately seven million sentence pairs, and use the GALE 2007 MT development set to tune the system, e.g. the model scaling factors, w.r.t. the BLEU score [17]. The GALE 2006 MT evaluation set is used as a blind test corpus.

For the source language, preprocessing consists of the steps described in Section 3 plus the removal of sentence-internal punctuation. For the target language, we mainly tokenize the corpora, i.e. separate punctuation marks from words. Additionally, we expand English contradictions like *it's* or *I'm* and remove the case information in order to reduce the vocabulary size and to improve the training. Regular expression are applied to categorize the corresponding numbers, URLs and e-mail addresses.

The automatic transcripts for the test sets are obtained using a system combination of three systems based on SRI's ASR system architecture.[2] For details about the ASR architecture the reader is referred to [18]. The Rover combination of the individual systems results in word error rates of 13.0% on the development set and 23.8% on the test set. More precisely, the combined system achieves error rates of 10.8% (BN) and 16.9% (BC) on the genre-specific parts of the development set.

6.2. Discussion

To measure the translation quality, we apply the automatic evaluation criteria also used in the official GALE evaluations, i.e. the BLEU score and the Translation Error Rate (TER) [19]. The BLEU score is the geometric mean of the n-gram precision in combination with a brevity penalty for too short sentences. TER measures the number of edits required to change a system output into one of the references. Both scores are computed case-sensitive w.r.t. a single reference translation.

As we lowercase the training corpus during preprocessing, we need to restore the correct case information. Therefore, we build a disambiguation language model. True-casing is done in a postprocessing step using the disambiguation tool from the SRILM toolkit. Compared to the correct case of the references, true-casing has an error rate of less than 2% on the dev set and about 3% on the test set.

Furthermore, we use the ICSI/UW algorithm to automatically segment the ASR transcripts into sentences. Obviously, this results in a sentence segmentation that is different from the segmentation of the reference translations. On document level, we align the system translations to the reference translations using our automatic sentence (re-)segmentation tool [20], which traces back the decisions of the Levenshtein edit distance algorithm.

The translation results for the GALE 2007 development set (DEV-BC/BN) and for the GALE 2006 evaluation set (TEST-BC/BN) are presented in Table 3. For the reranking experiments, we use the 10 000 best translation candidates.

Applying the domain adapted genre-specific LMs improves the system performance on the dev set as well as on the test set for both domains. The perplexity reductions reported in Section 4 hence also make a difference in translation quality. While we are able to further improve the scores

[2]We thank SRI International for providing us with the ASR transcripts.

System	Dev-BC		Dev-BN		Test-BC		Test-BN	
	BLEU [%]	TER [%]	BLEU [%]	TER [%]	BLEU [%]	TER [%]	BLEU [%]	TER [%]
GALE 2006	–	–	–	–	12.34	75.57	16.03	69.25
GALE 2007 (1st pass)								
- base	19.98	68.82	23.59	61.11	13.59	83.11	16.88	70.50
- dmix-gs	21.97	64.32	24.89	58.83	15.29	78.43	17.87	67.97
- dmix-gs*	22.84	64.19	25.64	58.39	14.99	79.23	18.24	68.29
GALE 2007 (2nd pass)								
- rescoring	23.50	62.77	27.00	56.91	15.16	78.43	18.53	67.00

Table 3. Translation results for the GALE 2007 development set (Dev-BC/BN) and the GALE 2006 evaluation set (Test-BC/BN); comparison to the RWTH system used in the official GALE 2006 evaluation and overview of current improvements.

Input	Dev-BC		Dev-BN		Test-BC		Test-BN	
	BLEU [%]	TER [%]	BLEU [%]	TER [%]	BLEU [%]	TER [%]	BLEU [%]	TER [%]
Correct Transcripts	27.90	53.62	28.79	53.27	22.05	58.72	20.72	62.13
Raw ASR Output	16.05	67.46	19.19	62.28	12.85	75.21	13.65	69.55
Normalized ASR Output	22.84	64.19	25.64	58.39	14.99	79.23	18.24	68.29

Table 4. Translation results for the GALE 2007 development set (Dev-BC/BN) and the GALE 2006 evaluation set (Test-BC/BN) for different inputs; correct transcripts vs. raw ASR output vs. normalized ASR output.

on the dev set by adding additional LM data and reranking the translation candidates, these improvements carry over only to the BN part of the test set. This is comprehensible because the additional LM data used in both passes are gathered only on news data. Furthermore the improvements due to the additional rescoring models are comparatively small as the test sets provide just single reference translations and therefore do not allow for a large tolerance in the MT output which can be exploited in the reranking pass.

To show the progress made, we compare the results with the official scores obtained in last year's GALE MT evaluation. W.r.t. to the BLEU score, already the baseline outperforms the 2006 system. This is due to the use of additional training data as well as new models, e.g. the phrase count features, and re-optimization of the entire system. Certainly, the advances of the ASR system account for better translations of the transcripts as well. Domain adapted LMs and the rescoring models further contribute to improve overall translation quality. We achieve improvements of 2.82% BLEU (BC) and 2.50% BLEU (BN) absolute. However, there are still shortcomings in our system. Regarding the TER scores, we only improve our system on the BN part of the test set. On the BC part, TER scores even deteriorate. We still have to analyze the translations in more detail but future steps require additional models and LM data that better match the BC domain.

Table 4 shows the results for different types of input. Given the correct transcripts, the system would be able to generate translations for the BC and BN sets that perform more or less at the same level. However, transcribing BC is a substantially harder task that is reflected in ASR error rates (10.8% on BN vs. 16.9% on BC for the dev set). Of course, this affects the MT performance. Translating automatically transcribed inputs, the BLEU scores drop from 28.79% to 25.64% on BN and from 27.90% to 22.84% on BC. Nonetheless, the numbers demonstrate how important the adjustment steps described in Section 3 are. The system performance shows clear deterioration for the translations of the unnormalized ASR output.

7. CONCLUSION

We have described the RWTH spoken language translation system that was used in the 2007 GALE Go/No-Go Translation Evaluation. The system uses a two pass approach; in the first pass, we use a dynamic programming beam search decoder to generate n-best translation candidates. In the second pass, these translations are reranked.

We have shown significant improvements compared to the GALE 2006 system achieved by introducing new feature functions based on phrase counts, applying domain adapted genre-specific language models, adding additional data and reranking the candidate translations. We have also described our work on adjusting the ASR and SMT vocabularies in a preprocessing step to MT and on predicting punctuation marks that are missing from automatically transcribed speech in the translation process.

Future work will focus on how to further adapt to domains that contain very noisy data and data being highly diverse from traditional newswire text, like broadcast conversations or web texts.

8. ACKNOWLEDGEMENTS

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA).

9. REFERENCES

[1] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA, July 2002, pp. 295–302.

[2] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, "A statistical approach to machine translation," *Computational Linguistics*, vol. 16, no. 2, pp. 79–85, June 1990.

[3] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, July 2003, pp. 160–167.

[4] E. Matusov, R. Zens, D. Vilar, A. Mauser, M. Popović, S. Hasan, and H. Ney, "The RWTH machine translation system," in *TC-STAR Workshop on Speech-to-Speech Translation*, Barcelona, Spain, June 2006.

[5] O. Bender, A. E. Isbihani, S. Hasan, S. Khadivi, J. Xu, R. Zens, Y. Zhang, and H. Ney, "The RWTH statistical machine translation system," in *Proceedings of the NIST Machine Translation Workshop*, Washington, DC, September 2006.

[6] A. Mauser, R. Zens, E. Matusov, S. Hasan, and H. Ney, "The RWTH statistical machine translation system for the IWSLT 2006 evaluation," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, Kyoto, Japan, November 2006.

[7] R. Zens, F. J. Och, and H. Ney, "Phrase-based statistical machine translation," in *25th German Conf. on Artificial Intelligence (KI2002)*, Aachen, Germany, September 2002, pp. 18–32, Springer Verlag.

[8] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proc. Int. Conf. on Spoken Language Processing*, Denver, CO, 2002, vol. 2, pp. 901–904.

[9] O. Bender, R. Zens, E. Matusov, and H. Ney, "Alignment Templates: the RWTH SMT System," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, Kyoto, Japan, September 2004, pp. 79–84.

[10] R. Zens and H. Ney, "N-gram posterior probabilities for statistical machine translation," in *Human Language Technology Conf. (HLT-NAACL): Proc. Workshop on Statistical Machine Translation*, New York City, NY, June 2006, pp. 72–77.

[11] A. E. Isbihani, S. Khadivi, O. Bender, and H. Ney, "Morpho-syntactic arabic preprocessing for arabic to english statistical machine translation," in *Human Language Technology Conf. (HLT-NAACL): Proc. Workshop on Statistical Machine Translation*, New York City, NY, June 2006, pp. 15–22.

[12] N. Habash and O. Rambow, "Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop," in *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Ann Arbor, MI, June 2005, pp. 573–580.

[13] N. Habash and F. Sadat, "Arabic preprocessing schemes for statistical machine translation," in *Proc. of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, New York City, NY, June 2006, pp. 49–52.

[14] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C++*, Cambridge University Press, Cambridge, UK, 2002.

[15] E. Matusov, D. Hillard, M. Magimai-Doss, D. Hakkani-Tur, M. Ostendorf, and H. Ney, "Improving speech translation with automatic boundary prediction," in *Proceedings of Interspeech 2007*, Antwerp, Belgium, August 2007, pp. 2449–2452.

[16] E. Matusov, A. Mauser, and H. Ney, "Automatic sentence segmentation and punctuation prediction for spoken language translation," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, Kyoto, Japan, November 2006, pp. 158–165.

[17] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA, July 2002, pp. 311–318.

[18] A. Stolcke, B. Chen, H. Franco, V. R. R. Gadde, M. Graciarena, M.-Y. Hwang, K. Kirchhoff, N. Morgan, X. Lin, T. Ng, M. Ostendorf, K. Snmez, A. Venkataraman, D. Vergyri, W. Wang, J. Zheng, , and Q. Zhu, "Recent innovations in speech-to-text transcription at SRI-ICSI-UW," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 1729–1744, September 2006.

[19] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proc. of the 7th Conf. of the Association for Machine Translation in the Americas (AMTA 06)*, Cambridge, MA, August 2006, pp. 223–231.

[20] E. Matusov, G. Leusch, O. Bender, and H. Ney, "Evaluating machine translation output with automatic sentence segmentation," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, Pittsburgh, PA, October 2005, pp. 148–154.