ISCA Archive
http://www.isca-speech.org/archive

Simultaneous German-English Lecture Translation

Muntsin Kolss[1], *Matthias Wölfel*[1], *Florian Kraft*[1], *Jan Niehues*[1], *Matthias Paulik*[1,2], *and Alex Waibel*[1,2]

[1]Fakultät für Informatik, Universität Karlsruhe (TH), Germany
[2]Language Technologies Institute, Carnegie Mellon University, USA
{kolss, wolfel, fkraft, jniehues, paulik, waibel}@ira.uka.de

Abstract

In an increasingly globalized world, situations in which people of different native tongues have to communicate with each other become more and more frequent. In many such situations, human interpreters are prohibitively expensive or simply not available. Automatic spoken language translation (SLT), as a cost-effective solution to this dilemma, has received increased attention in recent years. For a broad number of applications, including live SLT of lectures and oral presentations, these automatic systems should ideally operate in real time and with low latency. Large and highly specialized vocabularies as well as strong variations in speaking style – ranging from read speech to free presentations suffering from spontaneous events – make simultaneous SLT of lectures a challenging task.

This paper presents our progress in building a simultaneous German-English lecture translation system. We emphasize some of the challenges which are particular to this language pair and propose solutions to tackle some of the problems encountered.

1. Introduction

At educational institutions around the world, most lectures are given in the local language. Lectures and oral presentations form a core part of knowledge dissemination and communication, and as the number of exchange programs grows, this puts foreign students and visitors at a disadvantage and severely reduces opportunities for collaboration and exchange of ideas. Likewise, lectures and presentations at foreign organizations do not take place or are awkward because speaker and audience do not feel comfortable enough outside their native tongue. While human translation services are not an option to overcome the language barrier as costs would be prohibitive, effective automatic translation systems would go a long way towards making lectures and other presentations with live interaction more accessible.

Building an automatic system for simultaneous translation of lectures poses many challenges. First, lectures have a wide variety of topics that cover a virtually unlimited domain, and often go into much more detail than spoken language encountered in other speech translation tasks, such as limited-domain dialogs for travel assistance or translation of parliamentary speeches. The vocabulary and expressions can become very specialized with precise meanings that differ from the general usage. Since many lecturers are not professional speakers, the speaking styles can vary considerably and are much more conversational and informal than in prepared speeches or short utterances. The language is often ungrammatical and contains many disfluencies such as repetitions and false starts, and while the spoken input is usually a more or less constant flow of words, it does not necessarily consist of separable sentences with boundaries.

Obviously, such a system must run in real time since the speaker will not wait for the system to catch up. In addition, the system should translate with low latency, since a long delay between the original utterance, during which the speaker may navigate with a light-pointer over a slide, and the output of the translation will make it more difficult for the listener to follow the lecture, and could severely impact the understanding of the presented material.

In [1], a translation system for speeches and lectures was presented which translated English lectures into Spanish. Translating German lectures into English, as presented in this work, introduces additional difficulties not encountered in [1].

Speakers of non-English languages tend to embed more English words in their speech, especially for technical terms which are accepted in many disciplines. If unhandled, these English words will be misrecognized and cause additional follow-up errors. Similarly, the large number of compound words in the German language must be dealt with in both the speech recognizer, since they form an unlimited vocabulary, and the machine translation component, where alignment is based on word-to-word correspondences. The significant difference in word order between German and English poses another challenge for machine translation which is emphasized by the need for real-time processing and latency requirements.

In this paper, we present our progress in building a system for simultaneous translation of lectures from German to English. Section 2 gives an overview of the architecture and the components of our current system. In Sections 3 and 4, we explain in detail the speech recognition and machine translation components, respectively, and show individual results obtained during the development. Section 5 presents an end-to-end evaluation of the system and highlights a number of problems, and Section 6 concludes the paper.

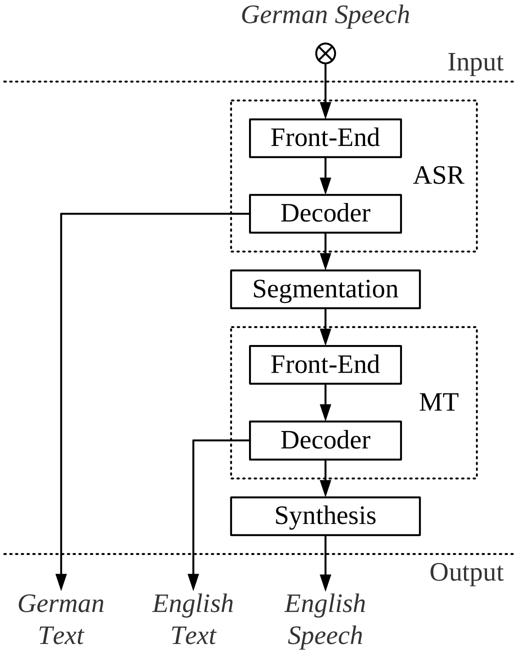

Figure 1: System overview of the individual components.

2. System overview

This section briefly describes the system overview of the lecture speech-to-speech translation system as summarized in Figure 1.

The *input* modality is limited to a single microphone. Additional input modalities could include additional microphones or slide information. The slide information can, for example, be used to adapt the language model weights on a slide by slide basis [2].

The speech-to-speech translation system can be decomposed into four main parts:

1. automatic speech recognition
2. segmentation
3. machine translation
4. speech synthesis

A detailed explanation of the different components follow in later sections. For synthesis we have used an English male voice provided by Cepstral [3].

The *output* modality contain the text transcripts in German, the written translation in English and a synthesized of the English text.

3. Speech recognition

In order to evaluate the speech recognition performance under realistic conditions we have recorded and transcribed 17 hours of German lecture speech (continuous, freely spoken) by native German speakers with different microphone types and speaker-to-microphone distances.

As a speech recognition engine we used the *Janus Recognition Toolkit* (JRTk). The speech recognition system configuration is described and individually evaluated in the next sections.

3.1. Front-end

Since it has been shown that *Mel frequency cepstral coefficients* (MFCC)s or *perceptual linear prediction* (PLP) coefficients are outperformed by warped MVDR cepstral coefficients [4] in noisy conditions, we decided to use the warped MVDR front-end for our experiments. The warped MVDR spectral envelopes, with a frame size of 16 ms and shift of 10 ms, were directly converted into a truncated cepstral sequence. The resulting 20 mean and normalized cepstral coefficients were stacked (seven adjacent left and right frames) and truncated to the final feature vector dimension of 42 by a multiplication with the optimal feature space matrix (the linear discriminant analysis matrix multiplied with the global semi-tied covariance transformation matrix [5]).

3.2. Acoustic model

The acoustic training material used for the experiments reported here consisted of approximately 17 hours of in-house recordings of lectures from various speakers, including the test speaker, resulting in 2,000 context dependent codebooks with up to 64 Gaussians with diagonal covariances each. The final acoustic models have been discriminatively trained due to maximum mutual information estimation where the speaker dependent adaptation matrices have been kept unchanged during training.

3.3. Vocabulary selection and compound splitting

The German language contains a large amount of compound words and morphological forms which cause additional difficulties for automatic transcription. *Compounds* are comprised of at least two nouns and thus an innumerable number of possible words can be formed. The number of *inflections* in German can be a multiple in comparison to the inflections in English. Thus the vocabulary size increases much faster for German than for English, causing either

- a bad vocabulary coverage (if truncated), or
- an underestimation of language model weights and
- a significant increase in search space complexity.

To reduce the number of words in the vocabulary of the recognition system and to reduce, at the same time, the *out of vocabulary* (OOV) words, we split the compound words into their constituents by comparison with a reference vocabulary. The reference vocabulary was created by *Hunspell* [6] using a base vocabulary tagged with word types and a corresponding affix list.

The final vocabulary set has been selected from small in-domain corpora (lecture transcriptions, presentation slides, web data) and large out-of-domain corpora (broadcast news). On the latter the *select-vocab* tool from the SRILM-toolkit [7], which estimates weighted and combined counts for all words, has been used to extract a ranked list of vocabulary entries. We reduced the resulting vocabulary size from approximately 255,000 entries to a manageable size of 65,000 words, to find a compromise between coverage and search space complexity. The reduced vocabulary size increased the OOV rate from 1.6% to 3.1%. We observed that this OOV rate is mainly due to special English expressions, compound words and inflection forms not seen in the training data. After merging the 65,000 words vocabulary with the vocabulary generated from the small in-domain corpora we were able to reduce the OOV rate to 2.3%. By splitting the compound words the OOV rate dropped to 1.5% (case sensitive) or 1.2% (case insensitive).

3.4. Language model

To train a 4-gram language model, we used the same small in-domain corpora and large out-of-domain corpora as has been used for vocabulary selection. For the web-collection we used the scripts, with small modifications to compensate for encoding issues arising for German text, provided by the University of Washington [8]. The search queries were created out of the most frequent bi- and trigrams from the in-domain corpora. The bi- and trigrams were kept if no stop-word was included and at least one word had been in upper case. We found that the casing restriction leads to better keywords since in German nouns have to be written in upper case. The queries consisting of a combination of 2004 bigrams and 768 trigrams retrieved approximately 10,000 HTML pages and 8,000 PDF files which resulted in 41 million words after appropriate filtering. The web-collection reduced the language model perplexity from 280 to 246 which significantly improved the word accuracy by more than 5.0% relative.

3.5. English words in German lectures

A general problem in automatic speech recognition is the transcription of foreign words. Our analysis of lectures in German language given at Universität Karlruhe (TH) showed that only 64% or 4195 out of 6589 words could be classified as uniquely *German*, i.e., words found only in the *German* Hunspell vocabulary. *English* words are represented by only 2% or 110 words in the transcripts, while 21% or 1397 words were represented in *both* languages. The remaining 887 words, most of them fillers, were found in neither the German nor the English Hunspell vocabulary and thus were counted as *unknown*.

An analysis of the recognition errors, see Table 1, shows that the English words cause a significant degradation in recognition performance. The overall word error rate of the evaluated system is 13.8%, but the German words have a word error rate of only 10.7% while the English words have a word error rate of 60.0%.

Language	German	English	Both	Unknown
Total Words	4195	110	1397	887
Deletions	52	1	44	0
Insertions	58	9	37	2
Substitutions				
German	258	37	91	113
English	7	6	8	7
Both	68	10	33	56
Unknown	5	3	2	4
Total Error	448	66	215	182
WER	10.7%	60.0%	15.4%	20.5%

Table 1: Absolute errors and word error rate (WER) by the different languages for the baseline system. Results from [9]

In order to reduce the performance gap between the German words in a German speech recognition system and the English words in a German speech recognition system, one could employ both the German as well as the English phoneme set (parallel) or map the English pronunciation dictionary to German phonemes (mapping). Both methods have been successfully applied as described in [9], and the best single method, mapping, leads to an absolute word error reduction by more than one percent.

3.6. ASR results

We evaluated on several different speaker-to-microphone distances, using unadapted (first pass) acoustic models, as well as acoustic models (second pass) after unsupervised adaptation with maximum likelihood linear regression (MLLR), constrained MLLR, and vocal tract length normalization.

We observe, Table 2, that the recognition accuracy suffers significantly by moving the microphone away from the speaker's mouth. The warped MVDR front-end is leading to better results than the MFCC front-end, in particular for severe acoustic environments. Different speech feature enhancement techniques to reduce the performance gap between distant and close recordings are investigated in [10].

We also compare the runtime performance of the system on different microphones which belong to different speaker-to-microphone distances respectively. The runtime factors have been measured on an Intel Xeon processor with 3.2 GHz and 3.5 GByte of RAM. We have observed that due to more difficult environment, the decoding in the ASR system takes longer for channels which are further away from the speaker's mouth. While no degradation in word accuracy can be observed between the CTM and the lapel microphone, the real time factor (RTF) increases by 25%. This increase is more severe for the table top and wall mounted microphone, 125% and 290% respectively.

Microphone	CTM		Lapel		Table Top		Wall	
Distance	5 cm		20 cm		150 cm		350 cm	
SNR	26 dB		23 dB		17 dB		12 dB	
Pass	1	2	1	2	1	2	1	2
Front-End	Word Error Rate %							
power spectrum	13.0	13.1	13.5	13.4	26.6	20.6	40.5	26.5
warped MVDR	13.4	13.4	13.1	13.1	26.8	19.7	39.5	24.7

Table 2: Word error rates for different speaker-to-microphone distances and front-ends.

4. Machine translation

In its baseline configuration, the machine translation component in our online lecture translator is a phrase-based statistical machine translation system which uses a log-linear combination of several models: an n-gram target language model, phrase translation models for both directions $p(f|e)$ and $p(e|f)$, a distance-based distortion model, a word penalty and a phrase penalty. The model scaling factors for these features are optimized on the development set by minimum error rate training (MERT) [11]. Search is performed using our in-house STTK-based beam search decoder which allows restricted word re-ordering within a local window during translation.

4.1. Word alignment

In statistical machine translation (SMT), parallel corpora are often the most important knowledge source. These corpora are usually aligned at the sentence level, so a word alignment is still needed. For a given source sentence f_1^J and a given target sentence e_1^I a set of links (j, i) has to be found, which describes which source word f_j is translated into which target word e_i.

Most SMT systems use the freely available GIZA++-Toolkit [12] to generate the word alignment. This toolkit implements the IBM- and HMM-models introduced in [13, 14]. They have the advantage of unsupervised training and are well suited for a noisy-channel approach, but introducing additional features into these models is difficult.

In contrast, we use the discriminative word alignment model presented in [15], which uses a conditional random field (CRF) to model the alignment matrix. This model is symmetric and no restrictions to the alignment are required. Furthermore, it is easy to use additional knowledge sources and the alignment can be biased towards precision or recall.

In the framework, the different aspects of the word alignment are modeled by three groups of features. The first group of features depend only on the source and target words and may therefore be called local features. The lexical features, which represent the lexical translation probability of the words as well as the source and target normalized translation probability of the words belong to this group. In addition, the following local features are used: The relative distance of the sentence positions of both words. The relative edit distance between source and target word is used to improve the alignment of cognates, and a feature indicating if source and target words are identical is calculated. Lastly, the links of the IBM4-alignments are used as an additional local feature. In the experiments this leads to 22 features.

The next group of features are the fertility features that model the probability that a word translates into one, two, three or more words, or does not have any translation at all. In this group there are indicator features for the different fertilities and a real-value feature, which can use the GIZA++ probabilities. To reduce the complexity of the calculation, this is only done up to a given maximal fertility N_f and there is an additional indicator feature for all fertilities larger than N_f. So 12 fertility features were used in the experiments.

The first-order features model the first-order dependencies between the different links. They are grouped into different directions. For example, the most common direction is $(1, 1)$, which describes the situation that if the words at positions j and i are aligned, also the immediate successor words in both sentences are aligned. In our configuration the directions $(1, 1)$, $(2, 1)$, $(1, 2)$, $(1, -1)$, $(1, 0)$ and $(0, 1)$ are used. For every direction, an indicator feature that both links are active.

Since the structure of the described CRF is quite complex, the inference cannot be done exactly. Instead, belief propagation was used to find the most probable alignment. The weights of the CRFs are trained using a gradient descent for a fixed number of iterations. The CRFs were first trained using the maximum log-likelihood criteria of the correct alignment. Since the hand-aligned data is annotated with sure and possible links, the CRFs is afterwards trained towards an approximations of the alignment error rate (AER).

4.2. Model training

The bulk of training data for our translation system comes from the parallel German-English *European Parliament Plenary Speeches* (EPPS) and *News Commentary* corpora, as provided by WMT08 [16]. To this we added a smaller corpus of about 660K running words consisting of spoken language expressions in the travel domain, and a corpus of about 100K running words of German lectures held at Universität Karlsruhe (TH) which were transcribed and translated into English, yielding a parallel training corpus of about 1.46M sentence pairs and 36M running words.

For machine translation experiments, we applied compound splitting to the German side of the corpus, using the frequency-based method described in [17] which was trained on the corpus itself. Even though this method splits more aggressively and generates shorter and partially erroneous word fragments, it produced better translation quality than the method used for the speech recognizer described in Section 3.3. After tokenization and compound splitting, we performed word alignment, using the GIZA++ toolkit and, for the final system, the method described in the previous section, and extracted bilingual phrase pairs with the Pharaoh toolkit [18]. 4-gram language models were used in all experiments, estimated with modified Kneser-Ney smoothing as implemented in the SRILM toolkit [7].

4.3. Model adaptation

The performance of data-driven MT systems depends heavily on a good match between training and test data in terms of domain coverage and speaking style. This applies even more so in our case, as university lectures can go deeply into a very narrow topic and, depending on the lecturer, are often given in a much more informal and conversational style than that found in prepared speeches or even written text.

To adapt the MT component towards this lecture style, we trained two separate sets of translation and language models: one set which was trained on the complete parallel training corpus for broad coverage, and additional smaller models trained only on the collected lecture data described in Section 4.2.

For decoding, we combine the two generated phrase tables into a single one, but retain disjunct model features for each part such that the corresponding feature scaling weights can be optimized separately during MERT. This approach improves significantly over the baseline of using a single phrase table trained on the complete corpus.

Likewise, our decoder uses the two language models with separate model scaling factors. For our system, this approach gives the same performance improvement as interpolating the language models based on perplexity on the development set. The improvements from model adaptation are shown in Table 3.

4.4. Front-end

In order to get the best translation results, the output of the speech recognizer has to be pre-processed to match the format expected by the translation component. We first perform compound splitting as described in Section 4.2, on top of the compound splitting already done in the speech recognizer.

Because the word order in German and English is very different, reordering over a rather limited distance like done in many phrase-based systems does not lead to a good translation quality. We experimented with rule-based reordering as proposed in [19], in which a word lattice is created as a pre-processing step to encode different reorderings and allow somewhat longer distance reordering. The rules to create the lattice are automatically learned from the corpus and the part-of-speech (POS) tags created by the TreeTagger [20]. In the training, POS based reordering patterns were extracted from word alignment information. The context in which a reordering pattern is seen was used as an additional feature. At decoding time, we build a lattice structure for each source utterance as input for our decoder, which contains reordering alternatives consistent with the previously extracted rules to avoid hard decisions.

	dev	test
Baseline	31.54	27.18
Language model (LM) adaptation	33.11	29.17
Translation model (TM) adaptation	33.09	30.46
LM and TM adaptation	34.00	30.94
+ Rule-based word reordering	34.59	31.38
+ Discriminative word alignment	35.24	31.40

Table 3: Translation performance in %BLEU on manual transcripts for model adaptation, rule-based reordering, and discriminative word alignment.

4.5. Input segmentation

Machine translation systems need a certain minimum context and perform best when given more or less well-formed sentence or utterance units to translate. The standard approach to coupling speech recognition and machine translation components into an integrated speech translation system is segmenting the unstructured first-best ASR hypothesis into shorter, sentence-like units prior to passing them to the subsequent machine translation component.

Choosing good segment boundaries, ideally corresponding to semantic or strong syntactic boundaries, can have a big impact on the final translation performance [21, 22, 23]. Some of the most useful features for segmentation are using a source language model, pause information from the original audio signal, since pauses often correspond to punctuation or semantic boundaries, and observing reordering and phrase boundaries within the translation system.

For simultaneous translation, we have the additional requirement that the segmenter, like the other components, must run in real-time. In our system, we use silence regions and a tri-gram source language model to identify segment boundaries, but apply additional thresholds to produce segments with an average length of about seven words. As Figure 2 shows, both the distribution of segment lengths for our automatic segmenter and manually produced segments also contain many segments which are considerably longer.

4.6. Stream decoding

The standard pipeline approach poses a dilemma for real-time speech translation systems: On the one hand, allowing longer segments generally leads to better translation quality. On the other hand, this translates directly into longer latency as well, i.e., the delay between words being uttered by the speaker and the translation of these words being delivered to the listener.

In addition, choosing meaningful segment boundaries is difficult and error-prone, and introducing boundaries has obvious drawbacks: each boundary destroys the available context for language modeling and finding longer phrase translations, and no word reordering across segment boundaries is possible. As we and others have found, rather long seg-

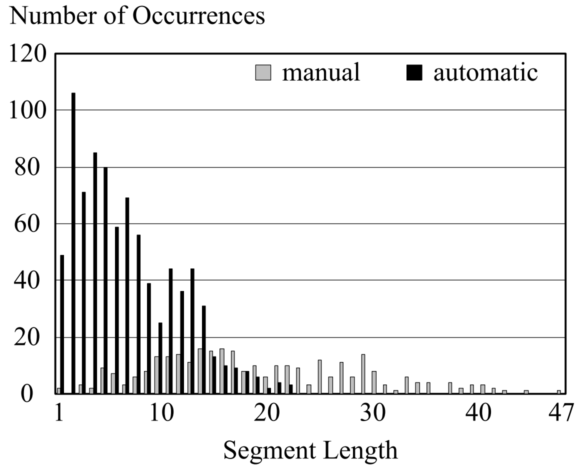

Figure 2: Number of occurrences of segment length for manual and automatic segmentation.

ment lengths of 10-40 words are necessary to get reasonably close to reaching the potential translation performance of the underlying translation system. A real-time system based on this approach must therefore compromise translation quality by using shorter segment lengths to avoid unacceptable latencies, and is still unable to achieve low latency because the drop in translation quality is quite severe as soon as this is enforced.

An alternative approach is completely by-passing the segmentation stage on the input side and directly processing the continuous input stream from the speech recognizer in real-time. The idea is to define a maximum delay or latency that we are willing to accept, and to decouple the decisions of when to generate translation output from any fixed input segmentation. Such a stream decoder design, proposed in [24], still allows for full word reordering within a sliding local window. Under such constraints, it outperforms the usual input segmentation pipeline and is able to reach its potential at very low latencies, making it especially well-suited for integrated real-time simultaneous translation systems.

4.6.1. Continuous translation lattice

Our baseline decoder creates one translation lattice from each input utterance or segment and adds all translation alternatives as additional edges. After the utterance has been decoded, this translation lattice is discarded for the next input utterance.

For stream decoding, in contrast, we maintain a continuous, rotating translation lattice to be able to process an ”infinite” input stream from the speech recognizer in real-time. New incoming source words from the recognizer are added to the end of the translation lattice, and the lattice is then immediately expanded with all newly matching word and phrase translation alternatives. When translation output has been generated for a part of the current translation lattice, the lattice is truncated at the start to reflect this.

4.6.2. Asynchronous input and output

Each incoming source word triggers a search for the best path through the current translation lattice, performed similarly as in the baseline decoder. However, the best translation hypothesis is not immediately output as translation. rather, the generation of output can be partially or completely delayed, either until a time out occurs, or until new input arrives from the recognizer, leading to lattice expansion and a new search.

This creates a sliding time window during which the translation output lags the corresponding incoming source stream, with the current translation lattice representing the still untranslated input.

Once the decision has been made to output a part of the best current translation hypothesis, the corresponding part of the translation lattice is removed from the start and the initial decoder hypothesis for the next search is set to the state at the end of the committed output.

4.6.3. Output segmentation

Two parameters are used by the stream decoder to decide which part of the current best translation hypothesis to output, if any at all:

- Minimum Latency L_{min}. The translation covering the most recent L_{min} untranslated source words received from the speech recognizer at any point is never output, except when a time out occurs. This effectively means that the decoder postpones translating the current end of the input stream until more context becomes available.
- Maximum Latency L_{max}. When the latency reaches L_{max} source words, translation output covering the source words exceeding this value is forced.

To find the output boundary, the currently best translation hypothesis is traversed backwards until the last L_{min} source words have been passed. If the hypothesis reached has no reordering gap, the translation up to this point is generated. Committing to a partial translation means that the lattice up to this point will be deleted, therefore some source words would remain untranslated if the hypothesis has some open reordering gaps.

If the hypothesis does contain open gaps, the traversal continues backwards until a state is reached where all word reorderings are closed. If no such state can be found, a new restricted search through the lattice is performed that only expands hypotheses which have no open reordering at the node where the maximum latency was exceeded, i.e. that have translated all source words up to that point.

4.6.4. Comparison with input segmentation

We compared the stream decoder with a baseline system which reached a performance of 29.04 %BLEU on manual

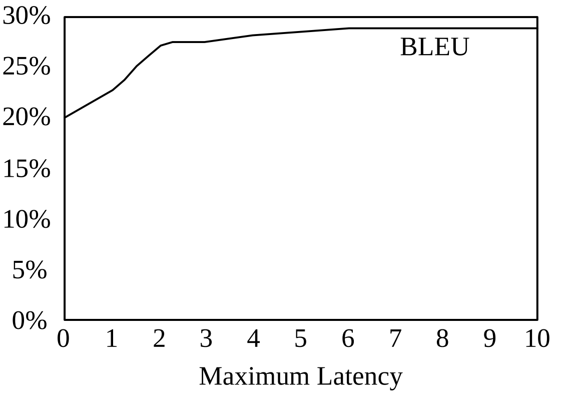

Figure 3: BLEU score vs maximum latency.

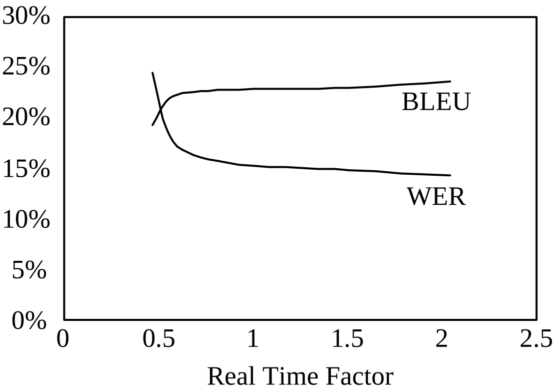

Figure 4: Real time factor vs. word error rate and BLEU score.

transcripts. Using the same models and local reordering window, the stream decoder was configured to respect a *maximum* latency of N words by setting the L_{max} parameter to the corresponding value. The L_{min} value was set to N/2, leading to an *average* latency of roughly N/2 words as well. As can be seen in Figure 3, the stream decoding approach performs as well as the baseline system at greatly reduced latency, and in addition is able to guarantee a strict maximum delay that will never be exceeded.

The quick saturation, however, also indicates that there is no potential for improvement without a much better model for long-distance reordering in the German-English language pair. Once such a model becomes available, the interaction between reordering and latency will have to be studied in greater detail.

5. End-to-end evaluation

We evaluated our system on a test set consisting of about 11,500 words from lectures held by two different native German speakers. Since our goal was to build a real-time system and the overall system speed is limited by the ASR component, we ran the system with different settings to come as close as possible to a setting which makes the most use of the available computation time. For these tests, we used input segmentation as described in Section 4.5 rather than the stream decoding described in Section 4.6.

5.1. Timing studies

Figure 4 plots the real time factor vs. word error rate and BLEU score. The different RTFs are due to a more or less restricted search beam in the speech recognition system. We observe a significant performance drop for RTF below 0.9. This threshold is determined by the acoustic environment and the number of words in the search vocabulary. For optimal real-time end-to-end performance, we tuned our vocabulary to reach a RTF of 0.9 on a lapel microphone with reasonable background noise.

5.2. Translation quality

Since translation quality scales almost linearly with the word error rate of the input hypotheses, good ASR performance is a prerequisite to producing good translations. For our system and task, a word error rate of about 15% marks the point where the translation output is becoming useful. As shown before, our system reaches this threshold under real-time conditions on standard hardware.

Recognition errors in the ASR output lead to a significantly lower translation performance of 23.65 BLEU, compared to 29.04 BLEU on the reference transcription. Nevertheless, the generated English online translation provides a good idea of what the German lecturer said. One of the most striking problems we found is the rather awkward word order produced by the online system, which strongly affects the readability of the English translation. For example, the system produced ”it is a joy here of course this talk to give” from the German input ”es ist eine Freude natürlich hier diesen Vortrag zu geben”, while the reference translation was ”it is a pleasure of course to give this talk here.” For this reason, our ongoing work focuses, among other things, on extending the rule-based word re-ordering described in Section 4.4. This approach allows us to apply longer distance reordering under the given real-time requirements.

6. Conclusion

We have presented our current system for simultaneous translation of German lectures into English, combining state-of-the-art ASR and SMT components. Several challenges specific to this task were identified and experiments performed to address them. The German language contains many compound words, and German lectures are often interspersed with English terms and expressions. Our ASR system was modified to better handle these points. Compound splitting was also used in the translation system to improve its quality. In addition, the different models of the translation systems were adapted to the topic and style of the lectures, and we experimented with reducing the latency of the real-time system. The different word order in the German and English language is still a challenge which will have to be better addressed in the future.

7. Acknowledgments

This work was partly supported by the Quaero Programme, funded by OSEO, French state agency for innovation. The authors would like to thank Susanne Burger, Oliver Langewitz, Edda Heeren, Stephan Settekorn and Marleen Stollen for transcribing and translating the lecture development and evaluation data, and Christian Fügen for useful discussions.

8. References

[1] C. Fügen, M. Kolss, D. Bernreuther, M. Paulik, S. Stüker, S. Vogel, and A. Waibel, "Open Domain Speech Recognition & Translation: Lectures and Speeches." in *Proc. of ICASSP*, Toulouse, France, 2006.

[2] T. Kawahara, Y. Nemoto, and Y. Akita, "Automatic Lecture Transcription by Exploiting Presentation Slide Information for Language Model Adaptation," *Proc. of ICASSP*, 2008.

[3] "Cepstral,"
http://www.cepstral.com/.

[4] M. Wölfel and J. McDonough, "Minimum Variance Distortionless Response Spectral Estimation," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 117–126, Sept. 2005.

[5] M. J. F. Gales, "Semi-Tied Covariance Matrices for Hidden Markov Models," *IEEE Transactions Speech and Audio Processing*, vol. 7, pp. 272–281, 1999.

[6] "Open Source Spell Checking, Stemming, Morphological Analysis and Generation." 2006,
http://hunspell.sourceforge.net.

[7] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *Proc. of ICSLP*, Denver, Colorado, USA, 2002.

[8] "Scripts for Web Data Collection provided by University of Washington,"
http://ssli.ee.washington.edu/projects/ears/WebData/
web_data_collection.html.

[9] S. Ochs, M. Wölfel, and S. Stüker, "Verbesserung der automatischen Transkription von englischen Wörtern in deutschen Vorlesungen," *Proc. of ESSV*, 2008.

[10] M. Wölfel, "A Joint Particle Filter and Multi-Step Linear Prediction Framework to Provide Enhanced Speech Features Prior to Automatic Recognition," in *Proc. of HSCMA*, 2008.

[11] F. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *41st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, 2003.

[12] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[13] P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer, "The Mathematics of Statistical Machine Translation: Parameter Estimation." *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[14] S. Vogel, H. Ney, and C. Tillmann, "HMM-based Word Alignment in Statistical Translation." in *COLING 96*, Copenhagen, 1996, pp. 836–841.

[15] J. Niehues and S. Vogel, "Discriminative Word Alignment via Alignment Matrix Modeling." in *Proc. of Third ACL Workshop on Statistical Machine Translation*, Columbus, USA, 2008.

[16] "Acl 2008 Third Workshop on Statistical Machine Translation,"
http://www.statmt.org/wmt08/.

[17] P. Koehn and K. Knight, "Empirical Methods for Compound Splitting," in *EACL*, Budapest, Hungary, 2003.

[18] P. Koehn and C. Monz, "Manual and Automatic Evaluation of Machine Translation between European Languages." in *NAACL 2006 Workshop on Statistical Machine Translation*, New York, USA, 2006.

[19] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *TMI*, Skövde, Sweden, 2007.

[20] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," in *International Conference on New Methods in Language Processing*, Manchester, UK, 1994.

[21] E. Matusov, D. Hillard, M. Magimai-Doss, D. Hakkani-Tür, M. Ostendorf, and H. Ney, "Improving Speech Translation by Automatic Boundary Prediction." in *Proc. of Interspeech*, Antwerp, Belgium, 2007.

[22] C. Fügen and M. Kolss, "The Influence of Utterance Chunking on Machine Translation Performance." in *Proc. of Interspeech*, Antwerp, Belgium, 2007.

[23] M. Paulik, S. Rao, I. Lane, S. Vogel, and T. Schultz, "Sentence Segmentation and Punctuation Recovery for Spoken Language Translation," in *ICASSP*, Las Vegas, Nevada, USA, April 2008.

[24] M. Kolss, S. Vogel, and A. Waibel, "Stream Decoding for Simultaneous Spoken Language Translation." in *Proc. of Interspeech*, Brisbane, Australia, 2008.