Towards Simultaneous Interpreting: The Timing of Incremental Machine Translation and Speech Synthesis

Timo Baumann[1], *Srinivas Bangalore*[2], *Julia Hirschberg*[3]

[1]Natural Language Systems Division, Department of Informatics, Universität Hamburg, Germany
[2]AT&T Labs – Research, Bedminster, USA
[3]Department of Computer Science, Columbia University, USA
mail@timobaumann.de, srini@research.att.com, julia@cs.columbia.edu

Abstract

In simultaneous interpreting, human experts incrementally construct and extend partial hypotheses about the source speaker's message, and start to verbalize a corresponding message in the target language, based on a partial translation – which may have to be corrected occasionally. They commence the target utterance in the hope that they will be able to finish understanding the source speaker's message and determine its translation in time for the unfolding delivery. Of course, both incremental understanding and translation by humans can be garden-pathed, although experts are able to optimize their delivery so as to balance the goals of minimal latency, translation quality and high speech fluency with few corrections. We investigate the temporal properties of both translation input and output to evaluate the tradeoff between low latency and translation quality. In addition, we estimate the improvements that can be gained with a tempo-elastic speech synthesizer.

1. Introduction

Today's speech-to-speech translation solutions are a long way from transparent and ubiquitous *universal translators* as envisioned in science fiction literature (e. g. [1]), for a multitude of reasons. One of the shortcomings is translation latency, which in speech can be described as the latency between when a concept can be grasped from listening to the source utterance and producing it as part of the target utterance. For swift and seamless communication across language barriers, low translation latency is key.

Incremental processing [2] is a technical means to implement interactive speech processing systems for online speech recognition [3], [4], [5], language understanding and generation [6], for speech synthesis [7]. Incremental processing has also been successfully applied to speech-to-speech translation (e. g. [8]), where it helps to bring down processing latency in an integrated system.

An important aspect of incremental processing (and hence, incremental translation) is the *granularity* at which material is being added. A fine granularity of processing is a precondition to low latency, as smaller units can more quickly be passed on to a next module. Previous work on incremental translation has focused on phrasing (based on intonation and somewhat related to meaning units) for translation [9], as phrases can easily be passed on to speech synthesis as one unit. Recently, incremental speech synthesis is progressing well at a word-by-word granularity, if some additional boundary and finality information is provided [10], [11].

In building language processing systems, joint analysis and optimization across module boundaries often greatly improves performance. The combination of speech recognition with understanding (e. g. [12]) or translation (e. g. [13]) is quite common, but this is less often done for the output side. (One notable exception is joint optimization of natural language generation and TTS [14], however not in an incremental setting.)

In this paper, we analyze the timing properties of source and target speech in an incremental machine translation setting in order to evaluate the improvements possible when combining word-by-word incremental machine translation with speech synthesis, particularly with respect to delivery latency. We do not yet actually employ fully incremental synthesis but focus our analysis on the advantages of such a synthesis technique in this contribution.

The remainder of this paper is structured as follows: in Section 2, we describe the interplay of incremental translation and the temporal unfolding of source and target speech based on an example and describe the basic strategies and evaluation metrics used in the study. In Section 3, we describe our corpus and experiment setup and present and discuss results for our basic strategies in Section 4. In Section 5, we look at advanced delivery timing that makes use of the flexibility that is made possible by incremental, just-in-time *tempo-elastic* speech synthesis. We summarize and conclude our work in Section 6 and outline future work in Section 7.

2. Timing Aspects of Simultaneous Interpreting

In a perfect world, a translator in transparent simultaneous interpreting will be able to come up with a perfect partial translation as soon as the corresponding source language word has

```
SRC: The | captain | waved | me | over | . |
TG1:       der/die*
TG2:              der Kapitän
TG3:              Der Kapitän winkte
TG4:              Der Kapitän winkte mich
TG5:              Der Kapitän winkte mich über*
TG6:              Der Kapitän winkte mich zu sich.
```

Figure 1: Depiction of successive incremental translation results (TGn) as words of the source utterance (SRC) are being processed. Wrongly translated words are marked by an asterisk(*). The challenge: given a (tokenized) input utterance, output should ideally commence immediately when correct translation results become available (but not before). Both source and target delivery durations must be taken into account.

been spoken by the source speaker.[1] Even in this case, the speech output component for incremental translation should consider when to start speaking rather than starting to speak immediately, as words in the target language may have a different duration than words in the source language; thus, the system could run out of words to speak, which would result in unnatural intermittent pauses during the utterance. Consider the example in Figure 1: here, even if the initial article is correctly translated to German "der", speech delivery should not commence immediately to avoid unnatural pauses if the next source language word might take longer to be uttered by the source speaker.[2] In Figure 1, translation output is purposefully aligned to show when respective words should ideally be delivered by synthesis in order to result in continuous speech output with minimal latency.

In an imperfect world, incremental translation will sometimes produce output that must later be revised (these words are marked with an asterisk in the figure; as luck has it, Google MT translates "the" to German "die", the female and plural form of the definite article, which turns out to be wrong in the example). Of course, a simultaneous translator should avoid speaking translations that later turn out to be wrong. Instead, it should speak with a high-enough latency to avoid short-range mistakes such as the ones shown in the figure.

Notice however, that the necessary delay to accommodate differences in delivery speed and intermittent translation errors can only be determined post-hoc, after the full utterance has been consumed. This of course defeats the goal of concurrent target language delivery.

We will present an analysis of the necessary delays per utterance under various translation conditions in Section 4. However, we believe that long-enough latency to account for all possible changes in translation cannot be the sole solution.

[1] Of course, our processing could also be concerned with sub-word units. However, that case would be conceptually similar to word-by-word processing (but potentially giving better results at the cost of higher complexity); this direction will not be considered further in the present work.

[2] This problem can be somewhat reduced by hesitation and/or lengthening capabilities: "de..r Kapitän ...").

Table 1: Some key statistics of the corpus (timings as determined by TTS; English reference data as well as token durations for *de*/*es* translations).

	count	duration in seconds		
	total	mean	stddev	median
utterances	1436	5.14	3.36	4.31
phrases (as determined by TTS)	3099	2.39	1.64	1.95
tokens	26890	.276	.172	.205
de token # and durations (in s)	27800	.328	.203	.25
es token # and durations (in s)	27275	.307	.195	.233

In order to account for long-range garden-pathing in translation (in which case translation *should* actively change its mind, just like a human in this situation), simply increasing delays is not the answer.

For this reason, we propose that automatic simultaneous interpreting modules, just like human experts, must have recovery capabilities, which enable them to cope with situations in which already-delivered parts of a translation should be revoked and replaced by a different translation. Human experts use and combine various strategies to cope with the problem [15]. We experiment here with the simplest possible solution of dealing with changes: we ignore all changes to words that have already or are currently being spoken. This causes the translation performance to deteriorate, given a fixed delay (similarly to [16]), which will also be analyzed in Section 4.

Finally, one intuitively important strategy of human experts is to vary the latency between input and output by varying speech delivery tempo. We report on our initial progress in determining overall latency and reducing it in Section 5.

3. Corpus and Experiment Setup

We use the IWSLT 2011 test set of the TED talks corpus as provided by the Web Inventory of Transcribed and Translated Talks [17]. As translation quality and stability may depend to a large extent on languages, we include analyses for three language pairs: $en \rightarrow de$, $en \rightarrow es$, and $de \rightarrow en$.[3]

We tokenize the respective source material with WASTE [18], using the included models for German and English. We then feed each of the utterances to standard, per-se non-incremental translation systems in a restart-incremental fashion: first translating just the first token, then the first two, then the first three, and so on, ending with the full utterance. This results in a large processing overhead and may confuse the translation system which may consider each input as a full utterance (while we are mostly sending partial utterances) – however it is a simple and reliable way of making non-incremental processors incremental. We decided to include all non-word tokens, as they give important clues to translation systems that are not trained on spoken data and are necessary to provide comparable BLEU score results on the TED data.

[3] Notice that we use the provided datasets 'in reverse' for $de \rightarrow en$ translation, ignoring the fact that that the original source becomes the target language in this experiment.

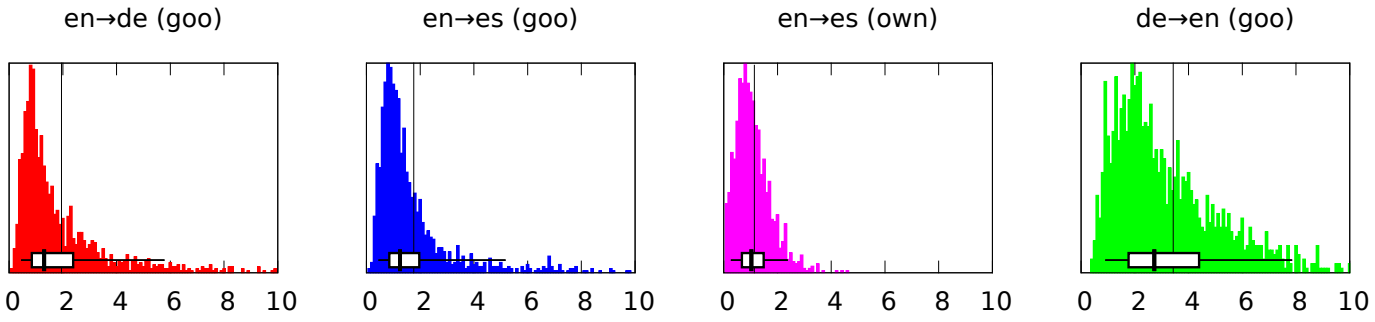

Figure 2: Histograms of per-sentence output delays (in s) that are necessary to accommodate all translation hypothesis changes.

As translation systems, we use both the Google MT web interface[4], and for Spanish, we also also use an AT&T proprietary SMT system [19].

Finally, we add word-level timing information to the source language and translation output using text-to-speech timing predictions[5] provided by MaryTTS [20] including recent additions for Spanish speech synthesis.[6] As our setup does not generate timings for some final punctuation, we use a flat duration estimate of 200 ms per punctuation in these cases. These 200 ms can be seen as the latency for end-of-utterance detection if our system were to be combined with incremental speech recognition in the future. Some key statistics about the corpus are compiled in Table 1.

As it turns out, overall German and Spanish speech duration are 23 % and 13 % higher respectively than overall English speech duration. A similar difference remains when using gold-standard German transcriptions instead of the MT output. Whether, however, this difference is due to a faster speech rate of the English voice, or due to expressive differences in the language, remains open. In any case, we have not controlled for this difference in the following experiments.

4. Evaluation of Basic Measures

For a time-aligned source sentence and its corresponding time-aligned incremental translation output that represents the final target language sentence, we find the minimum necessary delay at which the target sentence can be delivered such that the partial translation hypotheses always match the final target language sentence (i. e., the synthesis would never be triggered to start saying a word that is later replaced by a different word during incremental translation).

Using the incremental evaluation toolbox *intelida* [21], we compute the delay that is necessary in order to have all finally chosen target language words available before their scheduled delivery starts, and without intermittent interruptions from synthesis running out of words to speak. Delay histograms for all translation directions and systems are shown in Figure 2, and also indicate mean (vertical lines) as well as boxplots for median, 25/75 % (box) and 5/95 % (whiskers) quantiles. Notice that these delays are optimistic, i. e. they do not take into account translation time.

As can be seen in the histograms, the necessary delays are quite short on average, and, in particular, necessary delays for the majority of sentences are shorter than the average phrase length (cmp. Table 1), indicating that a word-level granularity (instead of phrase-level granularity as used in [9]) may be advantageous for simultaneous interpreting.

Also, we see that the histograms for the Google MT system have a very long tail with some necessary delays of over 10 seconds. On closer observation, we noticed that the Google MT system often (but not only) changes opinion when the final punctuation is added. We examined some of these sentence-final changes in detail and saw no clear tendency that they actually lead to an overall improvement of the resulting translation. In contrast, our own system, which is more strongly restricted in the sub-phrase reordering stage, results in a more normal distribution of necessary delays. This makes our own system more suitable for simultaneous interpreting, although the systems' translations and resulting BLEU scores differ, as shown below. Whether delay histograms would look more similar at equal BLEU performance levels must be left to speculation.

Finally, we notice much longer delays for $de \rightarrow en$ than for $en \rightarrow de$ translations. There may be several reasons for this: Firstly, German sentences often contain the verb late in the sentence, whereas English more stringently follows the SVO principle. As a consequence, the verb cannot be correctly translated until late in the sentence and, when it finally occurs, it may result in a change of the material that came before. Secondly, we mentioned above that our TTS generates slower speech for German (and Spanish) than for English. This phenomenon may skew the histograms in opposite directions when translating in opposite directions and may also be the cause for the longer necessary delays when translating from German. However, the histogram does not tail off as quickly

[4] `http://translate.google.com/` with the help of some PHP-based automation code.

[5] Of course, we could have extracted more precise source language timing information from TED videos, but results would likely be similar and only be available for English as source language.

[6] We thank Marcela Charfuelan for making a Spanish voice and linguistic resources available.

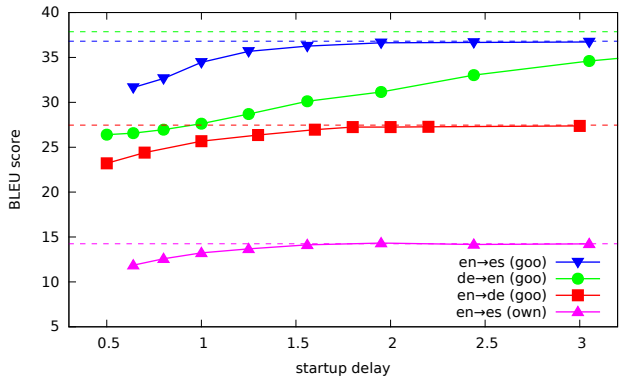

Figure 3: Performance penalty for given initial delays under all translation conditions.

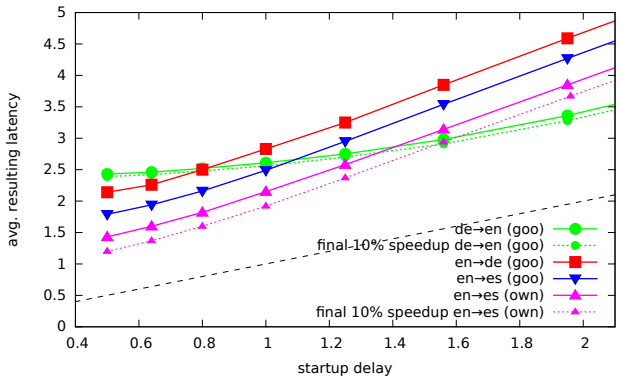

Figure 4: Resulting final latencies for given initial delays.

as it does with English as source language, which could not be explained by differing TTS tempos.

Of course, latency is just one aspect of simultaneous interpreting, the other major factor being translation quality. Beyond the non-incremental translation quality of the translation systems and corpora used, we have also implemented a very simple method for generating incremental translations under time-pressure (i.e., in simultaneous interpreting), where some words that are later overridden by a more informed translation, are already being spoken. In this case, our system simply ignores the change and re-aligns the new translation hypothesis using the Levenshtein algorithm [22].

Figure 3 shows translation performance (in terms of BLEU scores) of the different translation conditions for non-incremental (horizontal dashed lines) translation, which forms a natural upper bound for translations that are restricted in changing their hypotheses to different latency settings.

As can be seen in the figure, overall translation performance differs between translation systems, language pairs, and direction. Specifically, Google's $en \rightarrow de$ translations lags behind and differs substantially from the reverse translation direction, or $en \rightarrow es$. Our own $en \rightarrow es$ system performs poorly as compared to Google's. Our system was trained on different domain material, which may limit its performance on TED data; we plan to re-train our models in time for the final version of this paper.

Aside from translation quality, the performance penalty from limited-delay processing also differs substantially: our own system approaches its non-incremental performance rather quickly, while Google's systems require longer delays to reach their performance ceiling – although it must be noted that Google outperforms our system even with short delays.

Quite importantly, we note that $de \rightarrow en$ translation suffers most from long delays, to an extent that incremental performance is lower than $en \rightarrow es$, even though the non-incremental performance is higher, $de \rightarrow en$ only approaches non-incremental performance with a startup delay of around 4.8 seconds. We believe this property to stem from linguistic properties of German, which are not well-handled by our overly simplistic incrementalization approach.

5. Considering Speech Delivery Tempo

Words in the target language can only begin to be realized when the corresponding source language word has been completely delivered and translated. As words have different inherent durations, an incremental system may intermittently run out of material to speak, waiting for the next source language word to be completed and translated. As a consequence, the actual delay of the system as implemented is often higher than the initial startup delay, when the system has to wait for more translated material to become available. In addition, the speaking duration of the remaining words in the target language once the source utterance and translation have completed, must also be considered.

Figure 4 shows the actual resulting latency of target utterance completion after source utterance completion, at various initial delay settings. As can be seen in the figure, the resulting latency is substantially higher than the initial delay. There are two reasons for this: (a) target utterance delivery still needs to finish after the source utterance has already been completed; and (b) the delay may have to be increased intermittently, when source language delivery is too slow to sustain translation and delivery in the target language. In our experiments, we observe both phenomena, even though (a) is prevalent.

In all cases, latency is more than a second higher than the startup delay, which forms a natural lower bound for latency. We also notice that latencies increase more than initial delays for all but the $de \rightarrow en$ condition. This may be due to synthesis delivery speeds differing across languages. While $de \rightarrow en$ is impaired most by low delays (see Figure 3), it also accommodates longer delays in terms of the resulting latencies.

As mentioned previously, one goal of incremental speech synthesis is to have immediate control over delivery, and specifically, delivery timing. In Figure 4, we also plotted as dotted lines the resulting latencies if speech synthesis is sped up by 10 % starting at the word delivered after the source

language utterance has been completed. We notice a slight latency reduction across languages, of 4-5 % on average, or 100-200 ms, which may already be noticeable in applications.

We believe that somewhat higher speed-ups may be tolerable for listeners, which will lead to correspondingly larger improvements, and we plan to confirm this in listening/understanding experiments.

The second source of latency is delays that are increased during the utterance as the system runs out of target material. These additional delays can be substantial, especially for short startup delays. For example in the $en \rightarrow de$ condition, the additional average delay amounts to about 288 ms (179 ms) for a startup delay of 500 ms (respectively 700 ms).

We plan to reduce overall latency by bringing down utterance-internal delays through increasing speech tempo after the system has to intermittently pause. More generally speaking, we hope to estimate incremental translation stability (similarly to speech recognition stability [5]) and infer a flexible delay that accommodates more change at times when translation is particularly uncertain. The flexible delays will be integrated by varying delivery tempo in the incremental speech synthesis.

6. Conclusions

We have presented an analysis of incremental speech translation that takes into account speech delivery timings for both input and output. We find that, on average, conventional translation systems that are employed in a restart-incremental fashion produce their results with relatively low latencies. In particular, average delays are shorter than the average phrase, confirming our belief that word-by-word incrementality leads to better quality/latency trade-offs than phrase-by-phrase incremental systems.

In our experiments, we find that language pairs behave differently, and that German-to-English translation may be particularly difficult to perform incrementally. In addition, we find that our own system, which is quite limited in the word-reordering stage of translation, does not require as long delays and approaches its performance ceiling more quickly with limited delays – however, at the cost of overall lower performance. We plan to re-train our models with in-domain data in order to better compare our system with Google's MT.

In addition, we find that overall latency results from both the source utterance timing and its translation, and the target utterance delivery. While we have implemented a simple solution for the latter issue, we are still exploring how to deal with the former.

Finally, BLEU scores may be insufficient to judge incremental performance. An incremental translation system should strategically consider the duration of target language words in order to "gain time" or to speed up delivery, as required over the course of an utterance, while remaining easily understandable. Such word choices may hurt BLEU, as the "wrong" translation can be chosen, but improve actual system behaviour.

7. Future Work

As next steps, we will examine stability models for translations, similar to [5] for speech recognition. Our initial experiments in this direction are promising; however, they require translation internals which are not available from Google's MT. On the other hand, our own translation system is not trained on in-domain data, and hence delivers poor performance.

As we do not believe that a simultaneous interpreting system can lag behind to a degree that it "covers" all intermittent mis-translations, such a system will require an explicit recovery module that is able to rephrase and correct (perhaps using prosodic marking) already delivered material in a way that is easy to digest for the user. As such rephrasing cannot be learned from translation data, we believe this process cannot be left to the translation module alone.

Finally, we plan to validate the trade-off between translation quality and latency reduction of our system in a user study. In order to focus the study on the incremental aspects of the system, we plan to have participants fill in a multiple-choice survey about facts conveyed in the translation material. The timing of answers and their correctness will be informative regarding the two major aspects of incremental processing, latency and correctness. In addition, user changes to their answers should be useful in conveying information about the stability of the message conveyed.

8. Acknowledgements

The authors would like to thank Marcela Charfuelan for making available her MaryTTS extensions for Spanish speech synthesis, as well as the valuable feedback by the anonymous reviewers. This work is supported by a Daimler and Benz Foundation PostDoc Grant to the first author.

9. Bibliography

[1] D. Adams, *The Hitchhiker's Guide to the Galaxy*, ser. The Hitchhiker's Guide to the Galaxy. Pan Books, Oct. 1979.

[2] D. Schlangen and G. Skantze, "A General, Abstract Model of Incremental Dialogue Processing," in *Proceedings of the EACL*, Athens, Greece, 2009, pp. 710–718.

[3] T. Baumann, M. Atterer, and D. Schlangen, "Assessing and improving the performance of speech recognition for incremental systems," in *Proceedings of NAACL-HLT 2009*, Boulder, USA, 2009, pp. 380–388.

[4] E. Selfridge, I. Arizmendi, P. Heeman, and J. Williams, "Stability and accuracy in incremental speech recognition," in *Proceedings of the SIGDIAL 2011 Conference*, Portland, Oregon: Association for Computational Linguistics, Jun. 2011, pp. 110–119. [Online]. Available: `http://www.aclweb.org/anthology/W/W11/W11-2014`.

[5] I. McGraw and A. Gruenstein, "Estimating word-stability during incremental speech recognition," in *Proceedings of Interspeech*, ISCA, Portland, USA, Sep. 2012.

[6] G. Skantze and A. Hjalmarsson, "Towards incremental speech generation in dialogue systems," in *Proceedings of SIGdial*, Tokyo, Japan, Sep. 2010.

[7] T. Baumann and D. Schlangen, "INPRO_ISS: a component for just-in-time incremental speech synthesis," in *Procs. of ACL System Demonstrations*, Jeju, Korea, 2012.

[8] S. Bangalore, V. K. Rangarajan Sridhar, P. Kolan, L. Golipour, and A. Jimenez, "Real-time incremental speech-to-speech translation of dialogs," in *Proceedings of NAACL-HTL 2012*, Montréal, Canada, Jun. 2012, pp. 437–445.

[9] V. K. R. Sridhar, J. Chen, S. Bangalore, and A. Conkie, "Role of pausing in text-to-speech synthesis for simultaneous interpretation," in *Proceedings of SSW8*, 2013.

[10] T. Baumann, "Decision tree usage for incremental parametric speech synthesis," in *Proceedings of the International Conference on Audio, Speech, and Signal Processing (ICASSP 2014)*, Florence, Italy, May 2014.

[11] ——, "Partial representations improve the prosody of incremental speech synthesis," in *Proceedings of Interspeech*, 2014.

[12] A. Deoras, R. Sarikaya, G. Tur, and D. Hakkani-Tur, "Joint decoding for speech recognition and semantic tagging," Annual Conference of the International Speech Communication Association (Interspeech), 2012. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=183552.

[13] H. Ney, "Speech translation: coupling of recognition and translation," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, IEEE, vol. 1, 1999, pp. 517–520.

[14] C. Nakatsu and M. White, "Learning to say it well: reranking realizations by predicted synthesis quality," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia: Association for Computational Linguistics, 2006, pp. 1113–1120. DOI: 10.3115/1220175.1220315. [Online]. Available: http://www.aclweb.org/anthology/P06-1140.

[15] V. K. R. Sridhar, J. Chen, and S. Bangalore, "Corpus analysis of simultaneous interpretation data for improving real time speech translation.," in *INTERSPEECH*, 2013, pp. 3468–3472.

[16] H. Shimizu, G. Neubig, S. Sakti, T. Toda, and S. Nakamur, "Constructing a speech translation system using simultaneous interpretation data," in *Proceedings of the 10th International Workshop on Spoken Language Translation (IWSLT 2013)*, Heidelberg, Germany, 2013, pp. 212–218.

[17] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, 2012, pp. 261–268.

[18] B. Jurish and K.-M. Würzner, "Word and sentence tokenization with hidden markov models," *JLCL*, vol. 28, no. 2, pp. 61–83, 2013.

[19] V. kumar Rangarajan sridhar, S. Bangalore, A. Jimenez, L. Golipour, and P. Kolan, "SPECTRA: a speech-to-speech translation system in the cloud," IEEE International Conference on Emerging Signal Processing Applications, Tech. Rep., 2012.

[20] M. Schröder and J. Trouvain, "The German text-to-speech synthesis system MARY: a tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, no. 3, pp. 365–377, Oct. 2003, ISSN: 1572-8110. DOI: 10.1023/A:1025708916924.

[21] T. von der Malsburg, T. Baumann, and D. Schlangen, "TELIDA: A Package for Manipulation and Visualisation of Timed Linguistic Data," in *Proceedings of SigDial 2009*, London, UK, 2009.

[22] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics – Doklady*, vol. 10, no. 8, pp. 707–710, Feb. 1966.