IMPROVING ASR BY INTEGRATING LECTURE AUDIO AND SLIDES

João Miranda[1,2]*, João Paulo Neto*[1] *and Alan W Black*[2]

[1]INESC-ID / Instituto Superior Técnico, Lisboa, Portugal
[2]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
jrsm@l2f.inesc-id.pt, Joao.Neto@inesc-id.pt, awb@cs.cmu.edu

ABSTRACT

We propose a method to combine audio of a lecture with its supporting slides in order to improve automatic speech recognition performance. We view both the lecture speech and the slides as parallel streams which contain redundant information. We integrate both streams in order to bias the recognizer's language model towards the words in the slides, by first aligning the speech with the slide words, thus correcting errors on the ASR transcripts. We obtain a 5.9% relative WER improvement on a lecture test set, when compared to a speech recognition only system.

Index Terms— Lecture, Slides, Speech Recognition, System Combination

1. INTRODUCTION

With the proliferation of websites offering lectures, online courses, and conference talks, interest in the automatic transcription of such lectures has increased considerably. Automatic speech recognition of lectures enables several downstream applications such as machine translation of lectures into different languages, making them accessible on a larger scale and therefore increasing their usefulness.

Despite important advances in recent years, the performance of speech recognition systems can still deteriorate significantly when faced with adverse conditions such as reverberation, spontaneous speech which includes disfluencies, or word pronunciation mismatches in the case of non-native speakers or group-specific accents.

Currently, slides are often used as auxiliary materials to supplement a lecture or presentation. The aim of this paper is, therefore, to propose a method that combines lecture audio with slides in order to achieve improved speech recognition for these lectures. Our method relies on combining the lattices produced by a speech recognition system and those created from the slides, in order to extract a set of phrases towards which we bias the recognition results. The alignment produced by our technique also allows us to estimate the intervals of time during which each slide is being displayed by the speaker.

The remainder of this paper is organized as follows. Section 2 provides an overview of related work in the area. Section 3 summarizes the system that this work builds upon, our baseline system for combining multiple speech streams. Section 4 discusses the adaptations that were made to the multilingual speech stream algorithm in order to make it applicable to lecture transcription. Section 5 presents the results of the experiments designed to assess the algorithm's performance. Finally, Section 6 concludes and suggests ideas for future work.

2. RELATION TO PREVIOUS WORK

The work described in this paper is based on a parallel combination of speech and text streams with machine translation models. The integration of ASR and MT models has traditionally been carried out sequentially: in applications such as speech-to-speech or speech-to-text translation, the ASR outputs are handed over to the MT system, whether just the 1-best hypothesis produced by the recognizer or a set of alternatives (N-best lists, lattices or confusion networks). However, a number of researchers have realized the importance of integrating ASR or text with MT models in parallel, in order to take advantage of the redundant information available across these multiple streams. One can combine speech with a text stream, usually for an application such as machine-aided human translation [1, 2], in which a human translator dictates the translation, rather than typing it. Also, a few works have looked into combining several speech streams [3, 4], to improve ASR and MT systems in a simultaneous or consecutive interpretation scenario.

In previous work [5], we developed a system which combined the lattices generated by recognizers of original and interpreted speeches, in different languages, to yield improved recognition results. In order to link the language pairs together, we used phrase tables trained for a Statistical Machine Translation system. A sequence of words in the lattice of a given language is mapped to a corresponding sequence of words in the lattice of a different language through such a phrase table. The alignments that can be built from these correspondences allow one to uncover word sequences which originally had been assigned a low score by the recognizer,

but which are likely to have actually occurred in the speech, since the streams are assumed to be translations of each other. This process is described in Section 3. In [6], we also recover words that are not in the lattices produced by the recognizer, acronyms and pronunciations, using the redundancy provided by multiple streams. The current paper extends these works by integrating a new type of stream, which consists of slides, rather than speech, in the existing framework.

Previous work has also focused on the problem of automatically aligning speech and slides [7, 8, 9], for tasks such as multimedia indexing, retrieval or improved presentation. Often, these alignments are obtained through the use of similarity measures such as the cosine distance between the transcripts and the slides, or through dynamic programming algorithms [9]. Recently, the problem of correcting ASR transcripts using presentation slides has also been addressed [10]. In [10], the authors assume that the phonemes output by the ASR system are *noisy*, or distortions of the slide words, and that the true phoneme sequence is hidden. By performing inference on an HMM model with different states for slide and non-slide phonemes, and where the output distributions are modeled with a phoneme confusion matrix, they are able to recover from some of the errors produced by the recognizer. When compared to this prior work, our method has the advantage that it is more extensible: it can, in principle, use slides that are in a different language from the speech, and it allows speech to be combined not only with slides but also with other streams such as speech in a different language. It also does not require any initial alignment between speech and presentation slides, since that is built implicitly by the algorithm.

3. MULTISTREAM COMBINATION

Our baseline multistream combination method [5] takes multiple speech streams in different languages, as well as phrase tables that connect the language pairs for each combination of streams, and runs the following steps:

- Using a set of ASR systems (one for each language), transcribe the speech in each of the streams. We obtain a set of lattices that encode the different possibilities. In particular, we compute posterior probabilities for all n-grams with $n <= 3$.
- For each language pair, intersect the lattices with the respective phrase table, obtaining a set of bilingual phrase pairs that appear in both the lattices and the phrase table.
- Rescore the phrase pairs from the previous step, estimating their likelihood of actually having appeared in the speech. The highest-scoring among these pairs are used to construct a phrase pair alignment.
- The phrase pairs in the obtained alignment are used to rescore the lattices and generate new transcripts.

3.1. Intersection between lattices and phrase tables

The intersection step finds those phrase pairs $source \mid\mid\mid target$ that simultaneously are in the phrase table and for which both $source$ and $target$ can be found in the source and target lattices, respectively. The source and target phrases must be occur sufficiently close in terms of *time*. The maximum allowable time separation between the phrases is defined by an adjustable parameter δ, and phrase pairs not within this distance are not added to the intersection. The efficient computation of this intersection uses a specialized algorithm [5].

3.2. Phrase pair scoring and selection

Not all phrase pairs in the intersection are added to the output, since most of those appear by chance (very short, common words such as 'the' or 'a' occur very often in the lattices and are translations of each other). Instead, a number of features of each phrase pair are considered in scoring and selecting these to build an alignment, such as the posterior probabilities of each of the phrases in the phrase pair, its phrase table features, language model scores, and the time distances between both phrases of the pair. The output of this step is an alignment between phrase pairs, which consists of a set of non-overlapping phrase pairs with maximum total score.

3.3. Lattice rescoring

The rescoring step is an A* search of the lattices, producing new recognition hypotheses, where the language model is modified so that it assigns higher probability to word sequences that can be found in the generated alignments, at the correct times (i.e. whose time stamps match with the current time of the decoder).

4. ADAPTATIONS FOR LECTURE RECOGNITION

We adapted the method described in Section 3 for the purpose of lecture recognition. We considered the lecture speech to be one of the streams and the slides to be the other stream, and we dropped the initial decoding and rescoring steps for the slide stream. The input to our system consists of the lecture audio files and the slides in PDF format (which may contain slides from other lectures as well). Therefore, the slides have to be converted into lattices and we need to generate a phrase table connecting the two streams before our method can be applied.

4.1. Converting from slides to lattices

The slides are first pre-processed by extracting the unnormalized text from the slides, using the *pdftotext* tool. This text from the slides is then used to build a lattice which is used as input to the phrase table-lattice pair combination algorithm

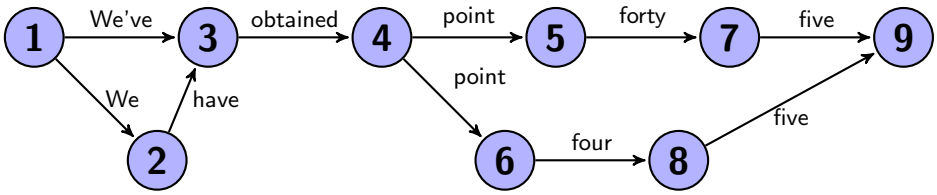

Fig. 1. Lattice generated for the sample sentence "We've obtained .45". The alternatives "we have" and "we've" have been generated for the first word, whereas "point four five" and "point forty five" were generated for the third word.

described in Section 3. To build the lattice, we keep a pointer to the most recently added node r. When processing a token t from the text, we create a node n and add an edge labeled t to the lattice, which connects r and n. Certain tokens, such as numbers, are spelled out as multiple words. For instance, *12000* is spelled as *twelve thousand*, so in this case an additional intermediate node is added to the lattice. Still other tokens have multiple possible normalizations, depending on the speaker and context. In an equation, for example, the token $<$ can be spelled as *lower than*, *smaller than*, or *less than*. Since we have no obvious way of choosing among them, we encode all of these as alternative paths between the nodes r and n. It is also possible to encode prior knowledge about which of the possible normalizations is more likely, by using different weights for different alternatives, although we did not do this in this paper. Figure 1 illustrates a lattice generated by this process for the sentence "We've obtained .45".

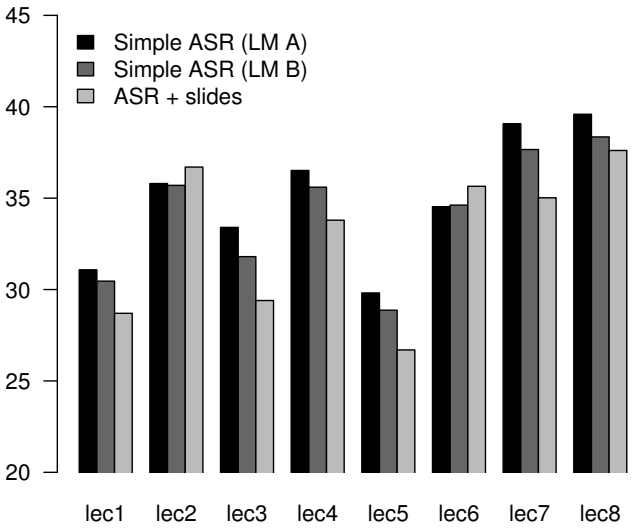

Fig. 2. Word Error Rate (WER) for each of the lectures in the test set, both when using speech recognition only (both with language models A and B) and when combining the speech with the presentation slides.

4.2. Modified alignment generation

As described in Section 3, our algorithm requests that we associate time stamps to each of the lattice nodes. However, the adaptation for lecture recognition requires that we relax this requirement since, unlike for speech streams, there is no time information directly associated with slides. Therefore, we modified our procedure so as to ignore the time differences involving a slide stream in the first iteration. In particular, no phrase pairs are discarded at this time due to large time differences. Then, at the end of the first iteration, we calculate the time stamps for the slides as follows: we take the speech-slide phrase pairs that were added to the alignment as *anchor points*, and to estimate time stamps for the remaining slide words and phrases, we linearly interpolate between the two closest such anchor points. We subsequently run a number of iterations of the algorithm, until the time stamps for the slide words converge or a predefined maximum number of iterations is reached.

4.3. Phrase Table generation

The obvious way of generating a phrase table to serve as input to our algorithm would be to create identical phrase pairs for all of the phrases in the slides with less than a fixed number of words, and then to add these phrase pairs to the phrase table. However, this ignores the fact that the lecturer will often substitute the words in the slides with morphologically related words. Generating all the morphological variants of a word is beyond the scope of the present work, so we attempt to generate only the most common among these. Our approach consists of first obtaining the part-of-speech and lemma for each of the words in the slides. Then, for a singular noun such as *probability* we include its plural *probabilities* as a translation of *probability* in the generated phrase table. Analogously, we include the singular form for plural nouns. Similarly, for a verb such as *to work*, we add the past participle form *worked* as well as the gerund *working*. If we encounter, for example, the gerund form, then we add the infinitive and past participle forms, to the generated phrase table, as possible translations.

5. RESULTS

In order to test the performance of our algorithm, we compared the baseline, which consists of speech recognition only, with the proposed method. We selected Audimus [11], a hybrid ANN-MLP WFST-based recognizer, as our baseline ASR system, and used our existing English acoustic models and lexica [12].

Our test set consisted of 8 lectures from the Stanford online Natural Language Processing course, together with the slides for each of the lectures. Four other lectures constituted a held-out development set used for parameter tuning. All of the lectures, both in the development and testing sets, were given by the same speaker. As reference transcripts for computing WER, we used the subtitles that were manually created for the course.

We then trained a domain-specific 4-gram language model using text extracted from a set of 10 computer science books. This language model was linearly interpolated with a 4-gram language model trained on the Hub4 text data, creating language model A, where the interpolation weight was estimated so as to optimize perplexity on our held out validation set. We also trained a language model (language model B) that included both the computer science books and the supporting slides for all of the lectures in the development and testing sets; again, we interpolated it with the 4-gram language model trained on the Hub4 data. Both language model estimation and interpolation were carried out using the SRILM toolkit [13].

For each of the talks in the testing set, we ran the baseline system with each of the two language models described, and then the system we developed to integrate the lecture speech with the slides. The results are summarized in Figure 2. We observe an overall improvement in results by using our method, from a baseline WER of 35% to 32.9% with an average relative WER improvement of 5.9%, when using language model A. When compared to the baseline system using language model B, our method has a smaller impact, as expected, but there is still a relative WER reduction of 3.6%. This result demonstrates that improving speech recognition with local information extracted from an alignment of speech and slides is more effective than simply interpolating the language model with text from the slides.

When considering the results at the individual talk level, however, there are significant variations. In talks 2 and 6, the results are slightly worse than the original ASR transcripts. We attribute this to differences between the lectures and their supporting slides: some of the slides contained less text or a larger number of images, which our text extraction method is not able to process, and in other cases the lecturer deviated from the slides to discuss a topic not covered by these. In those cases, our method seems to have introduced a small number of errors by trying to combine the speech with unrelated slide words.

6. CONCLUSIONS

In this paper we have described a technique to combine the speech from a lecture with the information contained in the slides used to support it in order to improve speech recognition performance. We achieved a 5.9% relative WER improvement over the baseline results (3.6% if the baseline language model is interpolated with the lecture slides).

In future work, we intend to apply the idea presented in this paper to cases in which the lecture and the slides are in different languages (e.g., the slides are in English but the lecture is given in Portuguese, or vice-versa). Additionally, we would like to combine these slides with other information streams that may be available in order to obtain improved performance. Finally, we want to explore alternative techniques for extracting text from slides, such as OCR, since that would enable us to access information not available to our current methods.

7. ACKNOWLEDGEMENTS

Support for this research was provided by the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the Carnegie Mellon Portugal Program under Grant SFRH/BD/33767/2009, and through projects CMU-PT/HuMach/0039/2008, CMU-PT/0005/2007, and PEst-OE/EEI/LA0021/2011.

8. REFERENCES

[1] S. Khadivi and H. Ney, "Integration of Speech Recognition and Machine Translation in Computer-Assisted Translation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1551–1564, 2008.

[2] A. Reddy and R.C Rose, "Integration of statistical models for dictation of document translations in a machine-aided human translation task," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2015–2027, 2010.

[3] M. Paulik, S. Stüker, C. Fügen, T. Schultz, T. Schaaf, and A. Waibel, "Speech Translation Enhanced Automatic Speech Recognition," in *Proceedings of the ASRU*, San Juan, Puerto Rico, 2005.

[4] M. Paulik and A. Waibel, "Extracting Clues from Human Interpreter Speech for Spoken Language Translation," in *Proceedings of ICASSP*, Las Vegas, USA, 2008.

[5] J. Miranda, J. P. Neto, and A. W Black, "Parallel combination of speech streams for improved ASR," in *Proceedings of the Interspeech*, Portland, USA, 2012.

[6] J. Miranda, J. P. Neto, and A. W Black, "Recovery of acronyms, out-of-lattice words and pronunciations from parallel multilingual speech," in *Proceedings of the SLT*, Miami, USA, 2012, to appear.

[7] D. Mekhaldi, "Multimodal document alignment: towards a fully-indexed multimedia archive," in *Proceedings of the Multimedia Information Retrieval Workshop, SIGIR*, Amsterdam, the Netherlands, 2007.

[8] G. Jones and R. Edens, "Automated alignment and annotation of audio-visual presentations," *Research and Advanced Technology for Digital Libraries*, pp. 187–196, 2002.

[9] W.T. Chu and H.Y. Chen, "Toward better retrieval and presentation by exploring cross-media correlations," *Multimedia systems*, vol. 10, no. 3, pp. 183–198, 2005.

[10] R. Swaminathan, M.E. Thompson, S. Fong, A. Efrat, A. Amir, and K. Barnard, "Improving and aligning speech with presentation slides," in *Proceedings of the International Conference on Pattern Recognition*, Istanbul, Turkey, 2010.

[11] H. Meinedo, D. A. Caseiro, J. P. Neto, and I. Trancoso, "AUDIMUS.media: a Broadcast News speech recognition system for the European Portuguese language," in *Proceedings of PROPOR*, Faro, Portugal, 2003.

[12] H. Meinedo, A. Abad, T. Pellegrini, I. Trancoso, and J. P. Neto, "The L2F Broadcast News Speech Recognition System," in *Proceedings of Fala2010*, Vigo, Spain, 2010.

[13] A. Stolcke, "SRILM-an extensible language modeling toolkit," in *Proceedings of the International Conference on Spoken Language Processing*, November 2002, pp. 257–286.