Exposé Bachelorarbeit - Better accuracy of automatic lecture transcriptions by using context information from slide contents

Jonathan Werner

Abstract

Automatic speech recognition (ASR) of university lectures is an important topic because of two main reasons: 1) providing accessibility for deaf people in the form of subtitles and 2) making the spoken text searchable, thus making it feasible to skim the content efficiently.

The main problem concerning ASR on academic lectures is the error rate for technical terms. This paper explores the possibility of improving recognition accuracy by adapting the language model with words that are extracted from the speakers slides.

Motivation

Scannability is crucial for academic research – but audio/audio-visual sources make this really hard right now. A concrete use case exists at the University of Hamburg with the *lecture2go* system which makes video recordings of many lectures available. It would be a great benefit for students that want to rewatch the lectures in preparation for an exam to have searchable transcriptions of the lectures. If the results of the augmented transcriptions are good enough they could be rather easily integrated into the *lecture2go* website and provide an actual benefit to students.

During a project at the University of Hamburg in the winter semester 14/15 we implemented a system which automatically generated subtitles for a given audio/video and a transcription. The system that I plan to implement for this thesis will integrate with this project. It would supply the missing first step in the toolchain: automatically generating the transcriptions (for the project we only used human generated transcriptions) with speech recognition. The software solution from our project could then align those results.

A live example of this idea can be seen at superlectures.com.[1] They present transcripts generated via automatic speech recognition, aligned to the video in a searchable text box beneath the video player. The main problem here is the bad recognition accuracy of technical terms, which diminishes the value of this solution: scanning the text is actually less effective than ‘scanning the video’ as the false-positives are confusing.

Problem space dimensions

There are three main dimensions of complexity in the problem space:

1. **Language model**: hotword-detection only versus *merging* of a domain-optimized general language model with talk-specific hotwords.

   Merging would be possible with CMU-Sphinx, but it is not clear if it’s possible to get access to such a domain-optimized model. The work presented in Cho et al. is very promising, but it seems that it is closed source only right now.

   The end result of hotword-only-detection would not be a complete transcript of the input but a time-indexed keyword-map. The use case of searching for technical terms could be still satisfied, however.

2. **Input format types and content**. I plan to start with latex sources which are available for selected talks. This can be extended in a second step to extract text from pdf slides. Additionally, more domain-specific data could be gathered from Wikipedia and Google search results. This is an approach that Kawahara, Nemoto, and Akita (2008) take. Comparing the results obtained with this additional data to those without is an interesting analytical opportunity.

3. **Local vs. global information**. In addition to specifying hotwords for the whole talk (‘global’), you can specify them to occur only during a given time interval (‘local’). This is an approach that Miranda, Neto, and Black (2013) explore.

   I will focus only on global information, as supplying local information implies associating the textual input tokens (which serve as the basis for hotword selection) with timing information. In the case of slide pdfs this would be essentially a optical character recognition (OCR) task, which is beyond the scope of this thesis.

[1] http://www.superlectures.com/sigdial2014/welcome-and-conference-overview-1

Work schedule

I will start by testing if and how the open source tool 'CMU Spinx' is able to adapt the recognition process in favor of a specific set of words.

I will then start implementing a prototype pipeline which extracts textual content from a set of input formats (for example latex sources, text extracted from slide PDFs) and feeds them into CMU Sphinx. This has the goal of finding out which parts would have to be manually implemented and which parts are already available. I would estimate that I need about one month of time to have a working software prototype.

I will then start measuring if the given process is able to increase recognition accuracy. As the primary evaluation criterium i will use a *keyword hit rate* metric. While this will require prior manual labeling of those keywords, the metric should be more meaningful compared to a generic word error rate.

I will compare the results and the evaluation criteria to the papers listed below. I would give this part another month. I would reserve the last month for polishing the general quality of the paper.

Paper structure

A first approach for the structure could look like this:

1. Introduction
    1. Comparison to related works
2. Pipeline Overview
3. Modules
4. Measurements, Analysis
5. Conclusion

I will start with a summary of the problem space and the proposed solution. I will summarize related work and the different approaches that the authors explored.

I will then give a high-level overview of my implementation, followed by a detailed explanation about the modules and their interaction. This will include an extensive part about the internal workings of the speech recognition process itself and comparisons and differences to the approaches outlined in the papers below.

I will then analyze the results and compare the recognition accuracy of the implementation with baseline measurements, using a *keyword hit rate* as a primary means of assessing the quality of the recognition.

Finally I will close with a summary and a conclusion about the value of the proposed solution.

Literature

I plan to use the following literature:

- Cerva et al. (2012)
- Maergner, Waibel, and Lane (2012)
- Miranda, Neto, and Black (2013)
- Kawahara, Nemoto, and Akita (2008)
- Cho et al. (2013)

The main focus in using this literature will be on distinguishing implementation differences concerning the ASR process (how exactly do they solve the problem of adapting/creating the language model), as well as comparing the results and the evaluation criteria.

Bibliography

Cerva, Petr, Jan Silovsky, Jindrich Zdansky, Ondrej Smola, Karel Blavka, Karel Palecek, Jan Nouza, and Jiri Malek. 2012. "Browsing, Indexing and Automatic Transcription of Lectures for Distance Learning." In *Multimedia Signal Processing (MMSP), 2012 IEEE 14th International Workshop on*, 198–202. IEEE.

Cho, Eunah, Christian Fügen, Teresa Herrmann, Kevin Kilgour, Mohammed Mediani, Christian Mohr, Jan Niehues, et al. 2013. "A Real-World System for Simultaneous Translation of German Lectures." In *INTERSPEECH*, 3473–77.

Cho, Eunah, Sarah Fünfer, Sebastian Stüker, and Alex Waibel. "A Corpus of Spontaneous Speech in Lectures: The KIT Lecture Corpus for Spoken Language Processing and Translation."

Kawahara, Tatsuya, Yusuke Nemoto, and Yuya Akita. 2008. "Automatic Lecture Transcription by Exploiting Presentation Slide Information for Language Model Adaptation." In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 4929–32. IEEE.

Maergner, Paul, Alex Waibel, and Ian Lane. 2012. "Unsupervised Vocabulary Selection for Real-Time Speech Recognition of Lectures." In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 4417–20. IEEE.

Miranda, Joao, Joao Paulo Neto, and Alan W Black. 2013. "Improving ASR by Integrating Lecture Audio and Slides." In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 8131–35. IEEE.