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

Jonathan Werner

Contents

Introduction

Scannability is crucial for academic research: you have to be able to quickly evaluate the usefulness of a given resource by skimming the content and looking for the parts that are specifically relevant to the task at hand.

The medium in which those resources are available is very centered on textual representation. Spoken content, hereinafter called *speech media* (audio- or audiovisual media that mainly consists of spoken language) doesn't make it possible to scan its contents. You are "stabbing in the dark" when looking for something specific in a medium like this and have to consume it like a linear narrative.

This means that although lectures and conference talks are a central element to science they are much more challenging and tedious to use for research work.

Being able to a) efficiently search and b) look at the temporal distribution of important keywords in a visually dense way would elevate the usefulness of speech media in the scientific context immensely.

One approach to accomplish those goals is utilizing Automatic Speech Recognition (ASR) to transcribe speech to text and also get timing information for the recognized words. This makes it possible to derive information about the density of given words at a given point of time in the talk, which in turn allows to compute word occurence density maxima. This opens up possibilities for compact visual representation of the interesting keywords, thus allowing the user to scan.

The main challenge when using ASR for this task is the recognition accuracy of technical terms. Most of them are not included in the language models that are available as those are broad and generic so as to optimize for accuracy over a wide topic spectrum. But when they are not included into the language model they have a very small chance to be correctly recognized at all.

So the usefulness of applying ASR with a generic language model to the problem is very small, as the intersection of interesting keywords with those technical terms that can not be recognized is very big.

The central goal of this thesis is to explore an approach to overcome this problem. This approach consists of using words from lecture slides or other notes to generate a lecture-specific language model. This is then interpolated with a generic language model and being compared to the 'baseline' accuracy of the generic model.

Structure of this thesis

The structure of this thesis is laid out as follows:

(1) **Research questions**

   I will state the research questions.

(2) **Scientific Background**

   (a) I will start by giving an overview over the state of the art of ASR and the most prevalent approaches.

   (b) I will explain the *concepts* which are fundamental for the understanding of speech recognition.

   (c) I will then examine the *scientific work* that has been done on applying ASR to the problem of lectures transcriptions.

   (d) Finally i will summarize the *metrics* that have been used to assess the quality of the improvements in different approaches.

(3) **Motivation**

   Here i will motivate why it is necessary to improve on the baseline performance of ASR in our context.

   I will talk about the role of keywords and technical terms and why they are not being detected and how that diminishes the usefulness of ASR for the purposes of scannability.

(4) **Test data**

   I will use the openly available *Open Yale Courses* [1], which provide a diverse selection of audio and video recordings of university lectures at Yale, additionally supplying quality manual transcriptions and course notes or slides.

   I will present the chosen courses, their selection criteria and discuss the range of types of lecture material.

(5) **The LM-Interpolation approach**

   (a) **Technical basis**

   I will introduce the open source speech recognition framework *Sphinx 4*. This is the software that is used for performing the actual recognition.

   (b) **Process overview**

   I will then give a overview of the design and architecture of our approach.

   (c) **Implementation**

   Finally i will describe the technical implementation by which the lecture material is compiled into a specialized language model and recognition is performed using a *interpolated* language model.

(6) **Analysis**

   (a) **Methods**
   I will discuss how to analyze the results and develop metrics that assess how well the given goals are met with our approach.

   (b) **Analysis**
   I will then perform quantitative analysis on our test dataset with the metrics we developed before.

   (c) **Discussion, Finding and Conclusions**
   I will discuss the findings and draw conclusions from the quantitative analysis concerning the effectiveness of our approach.

(7) **Visualization for Scannability**

I will present a prototype visualization method that uses the results from our approach to present a condensed representation of the keyword content from lectures with the goal of providing a quick, interactive way to search and scan speech media.

(8) **Improvements, Open Ends**

I will discuss possible improvements and open ends that were out of the scope of this thesis but would be interesting to explore further.

(9) **Summary**

I will end by summarizing the goals, the proposed approach, the design and implementation, the analysis and the results.

1 Research questions

The central research questions i want to investigate in this thesis can be formulated as follows:

(1) When we apply ASR to university lectures, what is the advantage of using an approach that consists of creating a lecture-specific language model and interpolating it with a generic language model, given that we are interested in improving the recognition accuracy of *interesting keywords* for the sake of searchability and scannability?

(2) What metric is useful for quantifying this advantage?

A secondary question is: How can we *use* the results from our approach to provide graphical interfaces for improving the users ability to search and scan the given speech medium?

The exploration of this question will not be the center of this thesis, but it will provide practical motivation for the results that the exploration.

2 Background

2.1 The field of Automatic Speech Recognition

Automatic Speech Recognition (ASR) can be defined as the process by which a computer maps an acoustic speech signal to text [2].

Rabiner and Juang [3] date the first research on ASR back to the early 1950s, when Bell Labs built a system for single-speaker digit recognition. Since then the field has seen three major approaches, which Marquard [4] summarizes as follows:

> 1. The *acoustic-phonetic approach* aimed to identify features of speech such as vowels directly through their acoustic properties, and from there build up words based on their constituent phonetic elements.
>
> 2. The *statistical pattern-recognition approach* measures features of the acoustic signal, and compares these to existing patterns established from a range of reference sources to produce similarity scores which may be used to establish the best match.
>
> 3. *Artificial intelligence (AI) approaches* have been used to integrate different types of knowledge sources (such as acoustic, lexical, syntactic, semantic and pragmatic knowledge) to influence the output from a pattern-recognition system to select the most likely match.

The most prevalent approach today is the *statistical pattern-recognition approach*, as it produces results with much higher accuracy compared to the acoustic-phonetic approach. The use of Hidden Markov Models (HMM) has been playing a key role in this approach, as it allows recognizers to use a statistical model of a given pattern rather than a fixed representation.

In the last years there has been a resurgence of AI approaches, specifically *deep learning approaches* [5]. The ASR paradigm we will use for this thesis will be limited to the former, however.

2.2 Dimensions of speech recognition

There are three dimensions which serve to classify different applications of speech recognition [2] [4]:

(1) **Dependent vs. independent**. Dependent recognition systems are developed to be used by one speaker, whereas independent systems are developed to be used by *any* speaker of a particular type, i.e North-American speakers. **Adaptive** systems lie between those poles, they are able to adapt to a particular speaker through training.

(2) **Small vs. large vocabulary**. Small vocabularies contain only up to a few hundred words and might be modeled by an explicit grammar, whereas large vocabularies contain tens of thousands of words so as to be able to model general purpose spoken language over a variety of domains.

(3) **Continuous vs. isolated speech**. Isolated speech consists of single words that are spoken with pauses in between them, whereas continuous speech consists of words that are spoken in a connected way. Continuous speech is significantly more difficult to recognize, as it is a) more difficult to find the start and end of words and b) the pronunciation of words changes in relation to their surrounding words.

With those three dimensions we can for example classify the application areas command and control systems, dictation and lecture transcription [4]:

Table 1: Three application areas

Application	Speaker	Vocabulary	Duration
Dictation	Dependent	Large	Connected
Command and control system	Independent	Small	Isolated
Lecture transcription	Independent	Large	Connected

The task of automatic lecture transcriptions can thus be characterized as speaker-independent (SI) large continuous speech recognition (LVCSR).

2.3 Concepts

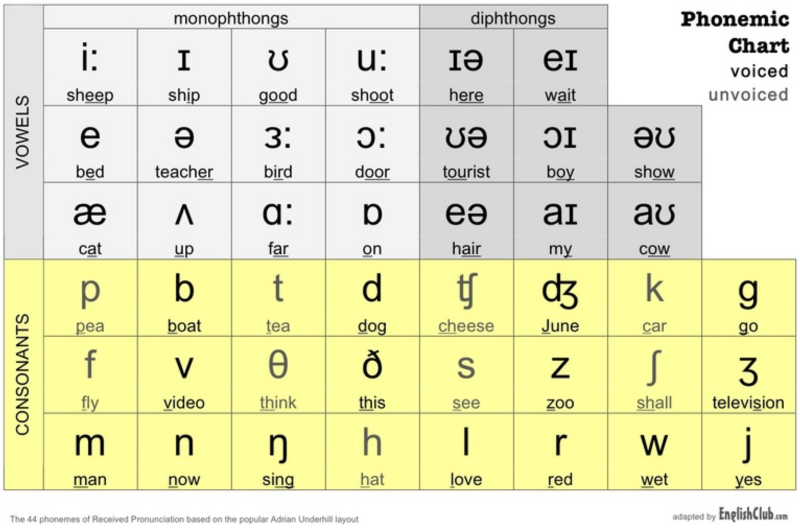
Speech recognition in the *statistical pattern-recognition approach* paradigm has three major concepts necessary for its understanding:

- phonemes and phonetic dictionaries
- acoustic models (AM)
- language models (LM)

2.3.1 Phonemes

A *phoneme* is “the smallest contrastive linguistic unit which may bring about a change of meaning” [6, p. 43]. They are the smallest unit of sound in speech which are combined to form words. The word *sun* for example can be represented by the phonemes `/s/`, `/u/` and `/n/`; the word *table* by `/t/`, `/a/` and `/bl/`.

A language together with a specific accent can be described by a set of phonemes that it consists of. Figure 1 uses symbols from the International Phonetic Alphabet (IPA) to display the 44 phonemes that are being used in Received Pronunciation (RP), which is regarded as the “standard accent” in the south of the United Kingdom [7].



	monophthongs				diphthongs			
VOWELS	i: sheep	ɪ ship	ʊ good	u: shoot	ɪə here	eɪ wait		
	e bed	ə teacher	ɜ: bird	ɔ: door	ʊə tourist	ɔɪ boy	əʊ show	
	æ cat	ʌ up	ɑ: far	ɒ on	eə hair	aɪ my	aʊ cow	
CONSONANTS	p pea	b boat	t tea	d dog	tʃ cheese	dʒ June	k car	g go
	f fly	v video	θ think	ð this	s see	z zoo	ʃ shall	ʒ television
	m man	n now	ŋ sing	h hat	l love	r red	w wet	j yes

Phonemic Chart
voiced
unvoiced

The 44 phonemes of Received Pronunciation based on the popular Adrian Underhill layout

adapted by EnglishClub.com

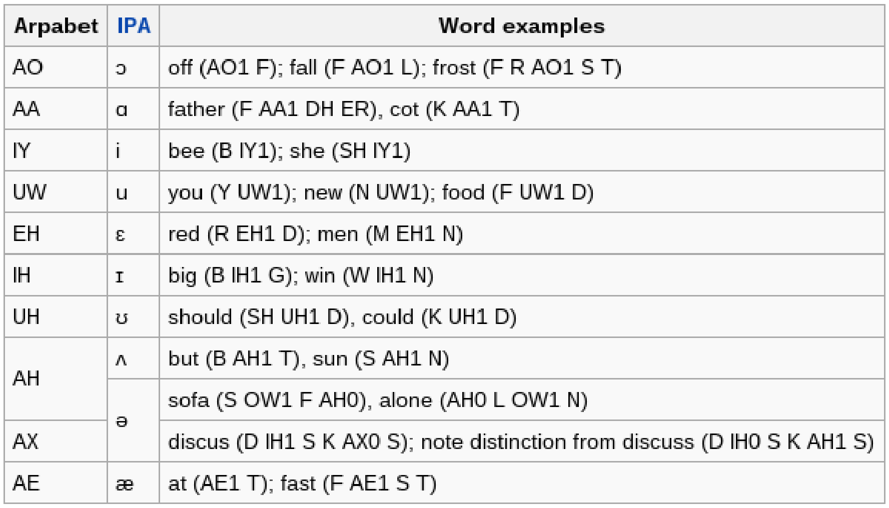
Figure 1: Phonemic Chart representing 44 phonemes used in RP British English

To be able to use phonemes in software an ASCII representation is more suitable. The standard for General American English is the *Arpabet*. Here each phoneme is mapped to one or two capital letters. The digits `0`, `1` and `2` signify stress markers: no stress, primary and secondary stress respectively. A comparison of the IPA format and the arphabet format can be seen in Figure 2, an excerpt that just shows the *monophthongs*.[1]

2.3.2 Phonetic dictionaries

Phonetic dictionaries map words to one or multiple versions of phoneme sequences.

[1] pure vowel sounds with relatively fixed articulation at the start and the end that don’t glide towards a new position of articulation



Arpabet	IPA	Word examples
AO	ɔ	off (AO1 F); fall (F AO1 L); frost (F R AO1 S T)
AA	ɑ	father (F AA1 DH ER), cot (K AA1 T)
IY	i	bee (B IY1); she (SH IY1)
UW	u	you (Y UW1); new (N UW1); food (F UW1 D)
EH	ɛ	red (R EH1 D); men (M EH1 N)
IH	ɪ	big (B IH1 G); win (W IH1 N)
UH	ʊ	should (SH UH1 D), could (K UH1 D)
AH	ʌ	but (B AH1 T), sun (S AH1 N)
	ə	sofa (S OW1 F AH0), alone (AH0 L OW1 N)
AX		discus (D IH1 S K AX0 S); note distinction from discuss (D IH0 S K AH1 S)
AE	æ	at (AE1 T); fast (F AE1 S T)

Figure 2: Excerpt from the Arpabet [8]

A phonetic representation of a word is specified manually from the knowledge how written words *actually sound* when spoken.

An excerpt from the dictionary `cmudict-en-us.dict` [9] looks like this:

```
...
abdollah AE B D AA L AH
abdomen AE B D OW M AH N
abdomen(2) AE B D AH M AH N
abdominal AE B D AA M AH N AH L
abdominal(2) AH B D AA M AH N AH L
...
```

The dictionary has 133.425 entries. In the general case only words that are in the phonetic dictionary being used can be recognized during speech recognition. *Grapheme*[2]*-to-Phoneme converters* (G2P) however make it possible to get phoneme sequence hypotheses for arbitrary words (that meaning arbitrary sequences of graphemes). While those results are on average less accurate than manually created variants, they play a vital role in texts with many technical terms as those are often not part of phonetic dictionaries.

[2] “The smallest unit used in describing the writing system of a language” [10], p.174

2.3.3 Acoustic models

An acoustic model (AM) describes the relation between an audio signal and the probability that this signal represents a given phoneme.

Acoustic models are created by *training* them on a *corpus* of audio recordings and matching transcripts. When being used in the context of speaker-independent recognition, those models are trained with a variety of speakers that represent a broad spectrum of the language/accent that the acoustic model should represent.

During the *decoding* phase the acoustic model and a phonetic dictionary are used to match sequences of small audio “slices” to possible phonemes and those phonemes to possible word sequence hypotheses.

However, acoustic models alone are not sufficient for speech recognition as they don’t have the “higher-level” linguistic information necessary to for example decide between homonyms and similar-sounding phrases such as “wreck a nice beach” and “recognize speech” [4, p. 11]. This information finally is provided by *language models*.

2.3.4 Language Models

Language models (LM) guide and constrain the search process a speech recognition system performs by assigning probabilities to sequences of words. They are trained by applying statistical methods on a text corpus. Analogous to acoustic models, generic language models use huge text corpora with a broad variety of topics. It is however possible to train language models on small and specialized text corpora, which is the central technical foundation for the approach discussed in this thesis.

The most commonly used form of language models are *n-gram language models*. In the context of a language model a *n-gram* is a sequence of n words. 1-grams are called *unigrams*, 2-grams are called *bigrams* and 3-grams are called *trigrams*. A *n-gram language model* maps a set of *n-grams* to probabilities that they occur in a given piece of text.

A key idea in modelling language like this is the *independence assumption*, which says that the probability of a given word is only dependent on the last n - 1 words. This assumption significantly decreases the statistical complexity and makes it thus computationally feasible.

N-gram language models don’t need to be constrained to one type of n-gram. The *Generic US English Generic Language Model* [11] from CMUSphinx we will use as the baseline for our approach for example consists of 1-, 2, and 3-grams.

A toy example of a language model with 1- and 2-grams when represented in *ARPA*-format (as used by CMUSphinx) looks like follows [12]:

```
\data\
ngram 1=7
ngram 2=7

\1-grams:
-1.0000 <UNK>   -0.2553
-98.9366 <s>     -0.3064
-1.0000 </s>     0.0000
-0.6990 wood     -0.2553
-0.6990 cindy   -0.2553
-0.6990 pittsburgh      -0.2553
-0.6990 jean     -0.1973

\2-grams:
-0.2553 <UNK> wood
-0.2553 <s> <UNK>
-0.2553 wood pittsburgh
-0.2553 cindy jean
-0.2553 pittsburgh cindy
-0.5563 jean </s>
-0.5563 jean wood

\end\
```

Here the first number in a row is the probability of the given n-gram in log_{10} format. This means that the unigram *wood* has a probability of $10^{-0.6990} \approx 0.2 = 20\%$ and the probability of the words “wood pittsburg” occuring in sequence is $10^{-0.2553} \approx 0.55 = 55\%$.

The optional third numeric column in a row is called *backoff weight*. Backoff weights make it possible to calculate n-grams that are not listed by applying the formula

```
P( word_N | word_{N-1}, word_{N-2}, ...., word_1 ) =
P( word_N | word_{N-1}, word_{N-2}, ...., word_2 ) *
  backoff-weight( word_{N-1} | word_{N-2}, ...., word_1 )
```

With the side condition that missing entries for `word_{N-1} | word_{N-2}, ...., word_1` are replaced by 1.0.

So if the text to be recognized would contain the sequence “wood cindy”, which does not appear as a bigram in the LM, the probability for this bigram could be calculated by `P(wood|cindy) = P(wood) * BWt(cindy)`.

Finally, the overall probability of a sentence with the words $w_1, ..., w_n$ can be approximated as follows:

$$P(w_1, ..., w_n) = \prod_{n=1}^{m} P(w_i \mid w_1, ...w_{i-1})$$

An example approximation with a bigram model for the sentence “I saw the red house” [13] represented as $P(\text{I, saw, the, red, house})$ would look like

$P(\text{I} \mid \langle s \rangle) \times P(\text{saw} \mid \text{I}) \times P(\text{the} \mid \text{saw}) \times P(\text{red} \mid \text{the}) \times P(\text{house} \mid \text{red}) \times P(\langle s \rangle \mid \text{house})$

2.4 Work done on ASR for lecture transcription

(1) I will then examine the *scientific work* that has been done on applying ASR to the problem of lectures transcriptions.

References

[1] *Open Yale Courses Website*. http://oyc.yale.edu/.

[2] *comp.speech Frequently Asked Questions*. http://www.speech.cs.cmu.edu/comp.speech/Section6/Q6.1.html.

[3] L. Rabiner and B.-H. Juang, "Fundamentals of speech recognition," 1993.

[4] S. Marquard, "Improving searchability of automatically transcribed lectures through dynamic language modelling," 2012.

[5] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and others, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[6] A. Cruttenden, *Gimson's pronunciation of English*. Routledge, 2014.

[7] A. Stevenson and M. Waite, *Concise Oxford English Dictionary: Book & CD-ROM Set*. Oxford University Press, 2011.

[8] *English Wikipedia Arpabet article*. https://en.wikipedia.org/wiki/Arpabet, 2015 (accessed 22.8.15).

[9] *CMU EN-US Pronouncing Dictionary (cmudict-en-us.dict)*. https://github.com/cmusphinx/sphinx4/blob/master/sphinx4-data/src/main/resources/edu/cmu/sphinx/models/en-us/cmudict-en-us.dict, 2015.

[10] C. Florian, "The Blackwell encyclopedia of writing systems," *Oxford: Blackwell*, 1996.

[11] *CMUSphinx US English Generic Language Model (cmusphinx-5.0-en-us.lm)*. http://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/US%20English%20Generic%20Language%20Model/, 2015.

[12] *CMUSphinx ARPA Language models*. http://cmusphinx.sourceforge.net/wiki/sphinx4:standardgrammarformats, 2015 (accessed 23.8.15).

[13] *English Wikipedia Language Model article*. https://en.wikipedia.org/wiki/Language_model, 2015 (accessed 23.8.15).