

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

Jonathan Werner

Contents

Introduction	3
Structure of this thesis	4
1 Research questions	6
2 Background	7
2.1 The field of Automatic Speech Recognition	7
2.2 Dimensions of speech recognition	7
2.3 Concepts	8
2.3.1 Phonemes	9
2.3.2 Phonetic dictionaries	10
2.3.3 Acoustic models	11
2.3.4 Language Models	11
2.4 Work done on ASR for lecture transcription	13
2.4.1 Generalization approaches	13
2.4.2 Specialization approaches	14
2.5 Metrics	14
3 Test data	14
3.1 Materials overview	15
3.1.1 Conclusion	16
4 The LM-Interpolation approach	17
4.1 Sphinx 4	17
4.2 Implementation	17
5 Analysis	23
References	23

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 occurrence 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* (*Open Yale Courses Website*, n.d.), 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 (*comp.speech Frequently Asked Questions*, n.d.).

Rabiner & Juang (1993) 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 (2012) 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* (Hinton et al., 2012). 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 (*comp.speech Frequently Asked Questions*, n.d., Marquard (2012)):

- (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 (Marquard, 2012):

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” (Cruttenden, 2014, 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 /bɪ/.

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 (Stevenson & Waite, 2011).

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

The 44 phonemes of Standard British English based on the popular Adon Underhill list

adapted by EnglishClub

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

¹pure 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	ə	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 (*English Wikipedia Arpabet article*, 2015 (accessed 22.8.15))

2.3.2 Phonetic dictionaries

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

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` (*CMU EN-US Pronouncing Dictionary* (`cmudict-en-us.dict`), 2015) 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. *Grapheme²-to-Phoneme converters* (G2P) however make it possible to

²“The smallest unit used in describing the writing system of a language” Florian (1996), p.174

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.

2.3.3 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/accents 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” (Marquard, 2012, 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 (CMUSphinx US English*

Generic Language Model (cmusphinx-5.0-en-us.lm), 2015) 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 (*CMUSphinx ARPA Language models*, 2015 (accessed 23.8.15)):

```
\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” occurring 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

$$P(\text{word}_N \mid \text{word}_{\{N-1\}}, \text{word}_{\{N-2\}}, \dots, \text{word}_1) = \\ P(\text{word}_N \mid \text{word}_{\{N-1\}}, \text{word}_{\{N-2\}}, \dots, \text{word}_2) * \\ \text{backoff-weight}(\text{word}_{\{N-1\}} \mid \text{word}_{\{N-2\}}, \dots, \text{word}_1)$$

With the side condition that missing entries for $\text{word}_{\{N-1\}} \mid \text{word}_{\{N-2\}}, \dots, \text{word}_1$ 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 $P(\text{wood}|\text{cindy}) = P(\text{wood}) * \text{Bwt}(\text{cindy})$.

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

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

An example approximation with a bigram model for the sentence “I saw the red house” (*English Wikipedia Language Model article*, 2015 (accessed 23.8.15)) represented as $P(\text{I, saw, the, red, house})$ would look like

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

2.4 Work done on ASR for lecture transcription

I will now give an overview over the scientific work done on lecture transcription, using Marquard (2012) as a guiding reference.

The research for speech recognition on lectures can be partitioned into three general approaches: generalization approaches, specialization approaches and approaches involving the user for manual correction and improvements.

2.4.1 Generalization approaches

Generalization approaches try to create models that capture common characteristics of lectures. Those characteristics include highly spontaneous presentation style and “strong coarticulation effects, non-grammatical constructions, hesitations, repetitions, and filled pauses” (Yamazaki, Iwano, Shinoda, Furui, & Yokota, 2007). Glass, Hazen, Hetherington, & Wang (2004) note the “colloquial nature” of lectures as well as the “poor planning at the sentence level [and] higher structural levels”.

The generalization approach has been applied on the acoustic model level: Cettolo, Brugnara, & Federico (2004) have examined adapting a generic acoustic model to account for spontaneous speech phenomena (“filler sounds”).

While the a subfield of ASR called “speaker diarization” tries to account for the interactivity between lecturers and students by identifying multiple speakers, most research treats lectures as single speaker events with the audience as background noise.

Generalization approaches at the language model level try to model common linguistic traits of the lecture genre (this can be called the “macro level”). Kato,

Nanjo, & Kawahara (2000) investigate topic-independent language modeling by creating a large corpus of text from lecture transcripts and panel discussions and then removing topic-specific keywords.³

2.4.2 Specialization approaches

Specialization approaches try to use context specific to a single lecture (“meso level”) or parts of a single lecture (“micro level”⁴).

Methods used for creating LMs from context information can be categorized into two approaches: direct usage of lecture slides and notes for the creation of LMs versus usage of “derived” data from those materials. Deriving data by using keywords found in slides, using them as web search query terms and using the found documents as the basis for LM creation is explored in Munteanu, Penn, & Baecker (2007), Kawahara, Nemoto, & Akita (2008) and Marquard (2012).

Using the whole text from lecture slides has been explored by Yamazaki et al. (2007). They compare the *meso level* with the *micro level* by dynamically adapting the LM for the speech corresponding to a particular slide. Kawahara et al. (2008) also examine dynamic local slide-by-slide adaption and compare it to global topic adaption using Probabilistic Latent Semantic Analysis (PLSA)⁵ and web text collection, concluding that the last performs worse than the former because of a worse orientation to topic words. .

2.5 Metrics

3 Test data

42 courses The test data i will use for evaluating our approach will be from *Open Yale Courses*⁶, which is a selection of openly available lectures from Yale university. It consists of 42 courses from 25 departments. Each course has about 20-25 sessions that have an average length of 50 minutes. Each lecture is provided with good quality audio and video recordings, precise manual transcripts and lecture material when available. Only about 20% of the lecture have lecture notes or slides at all and most materials from the natural and formal science departments (physics, astronomy, mathematics) consist of hand-written notes, making them unsuitable for our approach. All talks are in English.

³In a second step they combine this generalization technique with a specialization technique by adapting the resulting LM with a lecture-specific language model by using preprint papers of a given lecture.

⁴The three levels are taken from Marquard (2012).

⁵Latent Semantic Analysis is an approach to document comparison and retrieval which relies on a numeric analysis of word frequency and proximity.

⁶<http://oyc.yale.edu/>

I have chosen the following lectures: (Department, Course, Lecture Number - Title, abbreviation)

- *Biomedical Engineering*: Frontiers of Biomedical Engineering, 1 - What is Biomedical Engineering? (**biomed-eng-1**)
- *Environmental Studies*: Environmental Politics and Law, 8 - Chemically Dependent Agriculture (**environmental-8**)
- *Geology & Geophysics*: The atmosphere, the ocean, and environmental change, 8 - Horizontal transport (**geology-8**)
- *Philosophy*: Philosophy and the science of human nature, 8 - Flourishing and Detachment (**human-nature-8**)
- *Psychology*: Introduction to Psychology, 14 - What Motivates Us: Sex (**psy-14**)
- *Psychology*: Introduction to Psychology, 5 - What Is It Like to Be a Baby: The Development of Thought (**psy-5**)

The main selection criterion here was topical diversity, with the challenge that the majority of talks with computer-parsable notes was from the humanities.

3.1 Materials overview

The available material is very heterogeneous. I will now give an overview with excerpts which will serve as a basis for examining later if the quality and quantity of the supplied material is correlated with the amount of improvement of our approach.

geology-8 supplies a 2-page exercise sheet.

“Mars has a radius of 3.39×10^6 m and a surface gravity of 3.73 ms^{-2} . Calculate the escape velocity for Mars and the typical speed of a CO_2 molecule (assume $T = 250 \text{ K}$). How can Mars retain its CO_2 atmosphere? (Hint: the molecular weight of carbon dioxide is 44. Use the formulae given in class.) [...]”

biomed-eng-1 provides a 7-page glossary of technical terms.

“[...] active transport - the transport of molecules in an energetically unfavorable direction across a membrane coupled to the hydrolysis of ATP or other source of energy

ATP (adenosine 5'-triphosphate) - a nucleotide that is the most important molecule for capturing and transferring free energy in

cells. Hydrolysis of each of the two high-energy phosphoanhydride bonds in ATP is accompanied by a large free-energy change (“G”) of 7 kcal/mole

aquaporin – a water channel protein which allows water molecules to cross the cell membrane much more rapidly than through the phospholipid bilayer [...]”

human-nature-8 provides reading assignments for four books with short summaries each.

“[A] Epictetus, The Handbook

Background information about the Stoic philosopher Epictetus (c. 50-130 CE) and his famous work *Encheiridion* (The Handbook) appears in Nicholas White’s introduction to our translation. White has also added footnotes that explain points of potential confusion.

As the title indicates, The Handbook is intended as a tidy introduction to a more complex philosophical outlook. It is written in an accessible and engaging style.

The Stoic movement originated around 300 BCE and flourished for over five hundred years. The Stoics believed that the external world is deterministic: its state at any time is completely determined by its prior states. So, they maintained, it is pointless to wish for things to be different because to do so is to wish for something impossible. A wise person would, therefore, accept whatever befalls them without desiring that things go otherwise – hence the English word ‘stoic.’

Passages to focus on/passages to skim

I encourage you to read the text in full, at a steady reading pace. [...]”

psy-14/5 and **enviromental-8** provide ~10-page slides with a typical amount of text.

3.1.1 Conclusion

Only about 20% of the courses have lecture material at all; only about 20% of those actually have typical “slides” – the rest provides heterogenous other kinds of material. While it can not be inferred from this dataset that this is a general condition, it nevertheless shows a clear “real-world” disadvantage of an approach only relying on those materials. We will look at the impact of the varying quality and quantity in the analysis later.

4 The LM-Interpolation approach

I will now describe the LM-Interpolation approach. The high level overview looks like the following: we will use the open source speech recognition framework Sphinx 4⁷ as the software for performing speech recognition. Sphinx 4 has a modular architecture which allows specifying components of the whole process per configuration. It provides multiple implementations of LMs⁸, the default one being an ngram model.

It also provides an `InterpolatedLanguageModel`⁹ (ILM), which allows you to specify multiple LMs and weights and interpolate the probabilities for a given ngram from all models probabilities ($p = w_1 * p_1 + w_2 * p_2 + \dots$ where w_n are the weights ($\sum_{i=1}^n w_i = 1$) and p_i is the probability for a given ngram in LM_i).

The ILM's use in our approach is to factor in the importance of keywords. Those keywords have to be supplied in the form of an ngram language model. For this we extract text content from the lecture material, preprocess it and create an ngram LM from the resulting corpus. Sphinx4 is then a) run with a generic english ngram LM only and b) with the ILM configured to use the generic english LM and the keyword language model in a 50/50 weighting. Finally the two resulting transcriptions are compared with a selection of metrics.

As an example, the 1-gram *sex* has a probability of 2.82% in the keyword model from psy-14, but a probability of 0.012% in the generic english LM.¹⁰ When applying 50/50 interpolation, the result is $2.82\% * 0.5 + 0.012\% * 0.5 = 1.416\%$, which is an increase by the factor of ~117 over the generic probability.

4.1 Sphinx 4

4.2 Implementation

The pipeline is implemented with a collection of standalone command line tools and a set of bash and python scripts¹¹.

The tasks are the following, in chronological order:

1. Prepare the input

- The audio file is converted into Sphinx 4 compatible format (16khz, 16bit mono little-endian).

⁷Homepage: <http://cmusphinx.sourceforge.net/wiki/sphinx4:webhome>

⁸Overview: <http://cmusphinx.sourceforge.net/doc/sphinx4/edu/cmu/sphinx/linguist/language/ngram/LanguageModel.html>

⁹Javadoc: <http://cmusphinx.sourceforge.net/doc/sphinx4/edu/cmu/sphinx/linguist/language/ngram/InterpolatedLanguageModel.html>

¹⁰(*CMUSphinx US English Generic Language Model (cmusphinx-5.0-en-us.lm)*, 2015)

¹¹The source code is available here: <https://github.com/jonathanewerner/bachelor/tree/master/bin>.

- A testcase folder with a given shortname (e.g. `psy-15`) is created in the `results`-directory¹² of the source code repository.
- The reference transcript, the material (PDF format is required) and the converted audio file are moved into a `resources` subfolder of the testcase folder.

2. Create a keyword LM from lecture material

- `pdftohtml -i -xml` is applied on the given material PDF. The XML output representation is input to `pdfreflow`¹³. Compared to the tool `pdftotext` the combination of these 2 tools preserves paragraphs correctly, whereas `pdftotext` represents each line break in the input PDF as a new paragraph in the output text file. This is a significant disadvantage for the LM creation step, as a newline in the input file there has the semantic “end of sentence” – so that a sentence split into 4 lines by `pdftotext` would count as 4 sentences in the LM.
- The HTML output from `pdfreflow` is filtered by taking only relevant HTML-tags such as `<p>`’s (paragraphs) and `<blockquote>`’s, further improving the content-to-noise ratio.
- The resulting text is then preprocessed for optimal compatibility with the LM creation tool by removing punctuation and superfluous whitespace¹⁴.
- The resulting corpus is input to `estimate-ngram`, a LM creation tool from the MIT Language Modeling Toolkit¹⁵ (MITLMT).

3. Convert transcript to reference corpus

The transcript from Open Yale is supplied as HTML. We apply processing steps to transform it to a corpus ready to be consumed by the WER analysis tool (no punctuation, all lowercase). As these are specific to just the format chosen by Open Yale Courses, the details are omitted, as they have no general use.

4. Run Sphinx 4 in baseline and interpolated mode

`bin/sphinx-interpolated.py`¹⁶ supplies a wrapper for interfacing with Sphinx 4. The Java API of Sphinx 4 is exposed for command line usage by a JAR package which bundles the Sphinx 4 libraries and a small Main class. This class uses command line arguments supplied from `bin/sphinx-interpolated.py` to correctly configure Sphinx 4 and start the actual recognition.

Each testcase folder has a configuration file which specifies which models the test run should use:

¹²<https://github.com/jonathanewerner/bachelor/tree/master/results>

¹³`pdftohtml` and `pdfreflow` are open source linux command line utilities

¹⁴I use a combination of command line text processing (`sed`) and a perl script from Stephen Marquard here.

¹⁵<https://code.google.com/p/mitlm/wiki/EstimateNgram>

¹⁶<https://github.com/jonathanewerner/bachelor/blob/master/bin/sphinx-interpolated.py>

```

{
  "acousticModelPath": "en-new/cmuspinx-en-us-5.2",
  "dictionaryPath": "en-new/cmudict-en-us.dict",

  "languageModelPath": "en-new/cmuspinx-5.0-en-us.lm",
  "keywordModelPath": "model.lm",
  "g2pModelPath": "en-new/en_us_nostress/model.fst.ser",

  "resultsFolder": "biomed-eng-1"
}

```

`bin/sphinx-interpolated.py` interprets the “global” models relative to the repository root-folder `models`, the `resultsFolder` relative to the root folder `results` and the `keywordModelPath` relative to the `resultsFolder`. It then supplies the absolute paths to the JAR. It also supplies absolute output file paths for the transcription result and transcription word timings results.

This setup ensures reproducible results, as the environment of a given testcase is exactly specified (as long as the same binaries and script versions are assumed).

`bin/sphinx-interpolated.py` can now be used to run the baseline or/and interpolated version.

5. Analyze and compare the results

Finally the results from the two recognition runs are analyzed and compared by running `bin/hotword-analyze <testcase folder name>`. This performs two things: a) WER comparison and metrics generation and b) keyword visualization.

5.1 WER comparison and metrics generation

This first calls `bin/wer.py`¹⁷ on each run, which will calculate the WER and show a summary of substituted (SUB), inserted (INS) and deleted (DEL) words when comparing the reference (REF) to the hypothesis (HYP):

OP		REF		HYP
INS		****		this
INS		****		is
INS		****		that
OK		this		this
OK		is		is
OK		a		a
OK		course		course

¹⁷`wer.py` has been adapted from <http://progfruits.blogspot.de/2014/02/word-error-rate-wer-and-word.html>

```

SUB | a          | that
SUB | version    | aversion
OK  | of           | of
OK  | which        | which
SUB | i've         | i
OK  | taught       | taught
INS | ****         | him
...
...
{'Sub': 1230, 'Ins': 674, 'WER': 0.316, 'Del': 324, 'Cor': 5492}

```

In a second step it compares the two WER result files with `bin/compare-wer.py`.

The result is an HTML file with a) shows a WER comparison table and b) various statistical measures which will be explored later. The table (shown in Figure 3) colors correctly recognized words as green and incorrect words as red. It also marks words that have been improved in the interpolated version with a green border and words that have been worsened with a red border.

5.2 Keyword visualization

Data from the reference and the recognition results is compiled into a format suitable for consumption by a visualisation module, which will be discussed in depth further down in the chapter “Visualization for Scannability”.

All intermediate steps from the pipeline are represented as files in the testcase folder. Table 2 gives an overview of the files created by each pipeline step.

Table 2: File results of a testcase run

File	Description
Step 1: Prepare the input	
<code>resources/audio.mp3</code>	original audio
<code>resources/audio.wav</code>	converted audio
<code>resources/slides.pdf</code>	lecture material
<code>resources/transcript.html</code>	lecture transcript
<code>config.json</code>	run configuration
Step 2: Create a keyword LM	
<code>slides.corpus.txt</code>	lecture material corpus
<code>model.lm</code>	keyword LM
Step 3: Convert reference to corpus	
<code>reference.corpus.txt</code>	reference transcription corpus

File	Description
reference_wordcounts.json	reference transcription word counts ¹⁸
Step 4: Run Sphinx 4	
sphinx_log_baseline.txt	Sphinx 4 logging output
sphinx_log_interpolated.txt	
sphinx_result_baseline.txt	Sphinx 4 transcription
sphinx_result_interpolated.txt	
sphinx_word_times_baseline.txt	Sphinx 4 word times
sphinx_word_times_interpolated.txt	
Step 5.1: WER comparison / metrics generation	
results.json	run metrics in json format ¹⁹
wer_baseline.txt	WER table / metrics
wer_interpolated.txt	
wer_comparison.html	rich WER comparison + metrics
Step 5.2: Keyword visualization	
cloud_baseline.json	data representation for visualization
cloud_interpolated.json	

¹⁸They are needed for the visualization later.

¹⁹This eases parsability for aggregating multiple testcase results later.

Reference	baseline	interpolated
this	this	this
is	is	is
a	a	a
course	course	course
a	that	ab
version	aversion	version
of	of	of
which	which	which
i've	i	i
taught	taught	taught
almost	almost	almost
every	every	every
year	year	year
for	for	for
the	the	the
last	last	last
twenty	the	the
years	years	years
and	and	added
it	it	a

Figure 3: WER comparison

5 Analysis

References

- Cettolo, M., Brugnara, F., & Federico, M. (2004). Advances in the automatic transcription of lectures. In *Acoustics, speech, and signal processing, 2004. proceedings.(ICASSP'04). IEEE international conference on* (Vol. 1, pp. I-769). IEEE.
- CMU EN-US Pronouncing Dictionary (cmudict-en-us.dict)*. (2015). <https://github.com/cmusphinx/sphinx4/blob/master/sphinx4-data/src/main/resources/edu/cmu/sphinx/models/en-us/cmudict-en-us.dict>.
- CMUSphinx ARPA Language models*. (2015 (accessed 23.8.15)). <http://cmusphinx.sourceforge.net/wiki/sphinx4:standardgrammarformats>.
- CMUSphinx US English Generic Language Model (cmusphinx-5.0-en-us.lm)*. (2015). <http://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/US%20English%20Generic%20Language%20Model/>.
- comp.speech Frequently Asked Questions*. (n.d.). <http://www.speech.cs.cmu.edu/comp.speech/Section6/Q6.1.html>.
- Cruttenden, A. (2014). *Gimson's pronunciation of English*. Routledge.
- English Wikipedia Arpabet article*. (2015 (accessed 22.8.15)). <https://en.wikipedia.org/wiki/Arpabet>.
- English Wikipedia Language Model article*. (2015 (accessed 23.8.15)). https://en.wikipedia.org/wiki/Language_model.
- Florian, C. (1996). The Blackwell encyclopedia of writing systems. *Oxford: Blackwell*.
- Glass, J., Hazen, T. J., Hetherington, L., & Wang, C. (2004). Analysis and processing of lecture audio data: Preliminary investigations. In *Proceedings of the workshop on interdisciplinary approaches to speech indexing and retrieval at hLT-nAACL 2004* (pp. 9–12). Association for Computational Linguistics.
- Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6), 82–97. IEEE.
- Kato, K., Nanjo, H., & Kawahara, T. (2000). Automatic transcription of lecture speech using topic-independent language modeling. In *Sixth international conference on spoken language processing*.
- Kawahara, T., Nemoto, Y., & Akita, Y. (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 (pp. 4929–4932). IEEE.

Marquard, S. (2012). Improving searchability of automatically transcribed lectures through dynamic language modelling. University of Cape Town.

Munteanu, C., Penn, G., & Baecker, R. (2007). Web-based language modelling for automatic lecture transcription. In *INTERSPEECH* (pp. 2353–2356).

Open Yale Courses Website. (n.d.). <http://oyc.yale.edu/>.

Rabiner, L., & Juang, B.-H. (1993). Fundamentals of speech recognition. Prentice hall.

Stevenson, A., & Waite, M. (2011). *Concise Oxford English Dictionary: Book & CD-ROM Set*. Oxford University Press.

Yamazaki, H., Iwano, K., Shinoda, K., Furui, S., & Yokota, H. (2007). Dynamic language model adaptation using presentation slides for lecture speech recognition. *Proc. INTERSPEECH 2007*, 2349–2352.