IMPROVING SEARCHABILITY OF AUTOMATICALLY TRANSCRIBED LECTURES THROUGH DYNAMIC LANGUAGE MODELLING

by
Stephen Marquard

supervised by
Dr Audrey Mbogho

DISSERTATION SUBMITTED FOR THE PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF PHILOSOPHY IN INFORMATION TECHNOLOGY
IN THE DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF CAPE TOWN

December 2012

Abstract

Recording university lectures through lecture capture systems is increasingly common. However, a single continuous audio recording is often unhelpful for users, who may wish to navigate quickly to a particular part of a lecture, or locate a specific lecture within a set of recordings.

A transcript of the recording can enable faster navigation and searching. Automatic speech recognition (ASR) technologies may be used to create automated transcripts, to avoid the significant time and cost involved in manual transcription.

Low accuracy of ASR-generated transcripts may however limit their usefulness. In particular, ASR systems optimized for general speech recognition may not recognize the many technical or discipline-specific words occurring in university lectures. To improve the usefulness of ASR transcripts for the purposes of information retrieval (search) and navigating within recordings, the lexicon and language model used by the ASR engine may be dynamically adapted for the topic of each lecture.

A prototype is presented which uses the English Wikipedia as a semantically dense, large language corpus to generate a custom lexicon and language model for each lecture from a small set of keywords. Two strategies for extracting a topic-specific subset of Wikipedia articles are investigated: a naïve crawler which follows all article links from a set of seed articles produced by a Wikipedia search from the initial keywords, and a refinement which follows only links to articles sufficiently similar to the parent article. Pair-wise article similarity is computed from a pre-computed vector space model of Wikipedia article term scores generated using latent semantic indexing.

The CMU Sphinx4 ASR engine is used to generate transcripts from thirteen recorded lectures from Open Yale Courses, using the English HUB4 language model as a reference and the two topic-specific language models generated for each lecture from Wikipedia.

Three standard metrics – Perplexity, Word Error Rate and Word Correct Rate – are used to evaluate the extent to which the adapted language models improve the searchability of the resulting transcripts, and in particular improve the recognition of specialist words. Ranked Word Correct Rate is proposed as a new metric better aligned with the goals of improving transcript searchability and specialist word recognition.

Analysis of recognition performance shows that the language models derived using the similarity-based Wikipedia crawler outperform models created using the naïve crawler, and that transcripts using similarity-based language models have better perplexity and Ranked Word Correct Rate scores than those created using the HUB4 language model, but worse Word Error Rates.

It is concluded that English Wikipedia may successfully be used as a language resource for unsupervised topic adaptation of language models to improve recognition performance for better searchability of lecture recording transcripts, although possibly at the expense of other attributes such as readability.

Acknowledgements

I extend my thanks to:

- My supervisor, Dr Audrey Mbogho, for her patience and wise counsel.
- My wife, Pippa, and son, Cael, for their support.
- Developers of the open source toolkits used for this project, in particular Nickolay V. Shmyrev (CMU Sphinx4), Radim Řehůřek (gensim) and Josef Novak (phonetisaurus), for sharing freely both their code and expertise.
- Timothy Carr and Andrew Lewis from ICTS, who liberated me from the constraints of desktop computing.

Computations were performed using facilities provided by the University of Cape Town's ICTS High Performance Computing team: http://hpc.uct.ac.za.

Copyright and License

Contents

Tables

Figures

1 Introduction

1.1 Lecture recording in universities

Lecture capture technologies are gaining popularity in higher education [1]. Such systems record audio, video and presentation slides or graphics from a lecture, so that the lecture can be played back later by students or the general public.

However, a single continuous recording is often unhelpful for users. As students often use lecture recordings for revision or preparation for assessments [2], they may wish to play back a part rather than the whole of a lecture, or identify lectures containing particular material or concepts. To support this, various indexing schemes have been used to enable faster navigation and searching. For example, slide images are commonly used to provide a visual index within the lecture (Figure 1-1).

Figure 1-1: A lecture recording showing slide thumbnails as a navigation aid

However, a transcript of the lecture provides even more possibilities, as it enables:

- quick navigation within the lecture
- discovery through text search across lectures within the lecture capture system
- for public lectures, discovery through search engines and content aggregators.

1.2 Integration of speech recognition into a lecture capture system

In many contexts, producing manual transcripts from audio recordings is not economically viable as it is time-consuming and expensive. Using automated speech recognition (ASR) technologies for transcription is thus an attractive lower-cost approach.

ASR may be integrated into an automated lecture capture system in the processing phase when recorded media are ingested and processed on a central cluster prior to distribution to end users. Media analysis tasks in the processing phase may include segmenting video into slides, optical character recognition (OCR) of text in slides, and speech recognition. Figure 1-2 shows the architecture of an open source lecture capture framework, Opencast Matterhorn [3], [4].

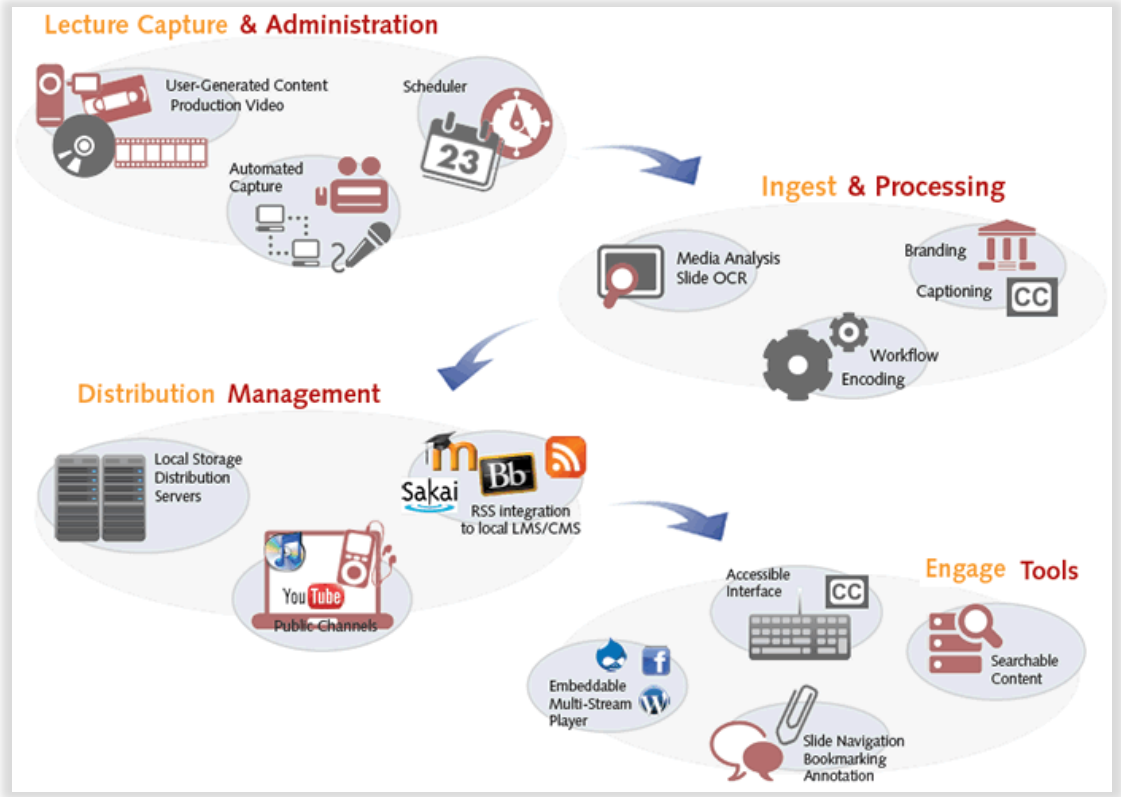


Figure 1-2: Architecture of the Opencast Matterhorn Lecture Capture System

The actual task of speech recognition and generating the transcript may be undertaken by a software component internal to the lecture capture system, or could be performed by an external service, for example provided by a vendor as a software-as-a-service (SaaS) offering.

The time-aligned transcript created by the ASR engine forms part of the recording metadata, and may be:

- exposed to the end-user through the playback user interface
- indexed within the capture system to enable searching across lectures
- exposed to external harvesters through RSS, OAI-PMH or other metadata feeds

An example of text-based search and navigation within a recorded lecture is shown in Figure 1-3 from the MIT Lecture Browser prototype [5], [6].

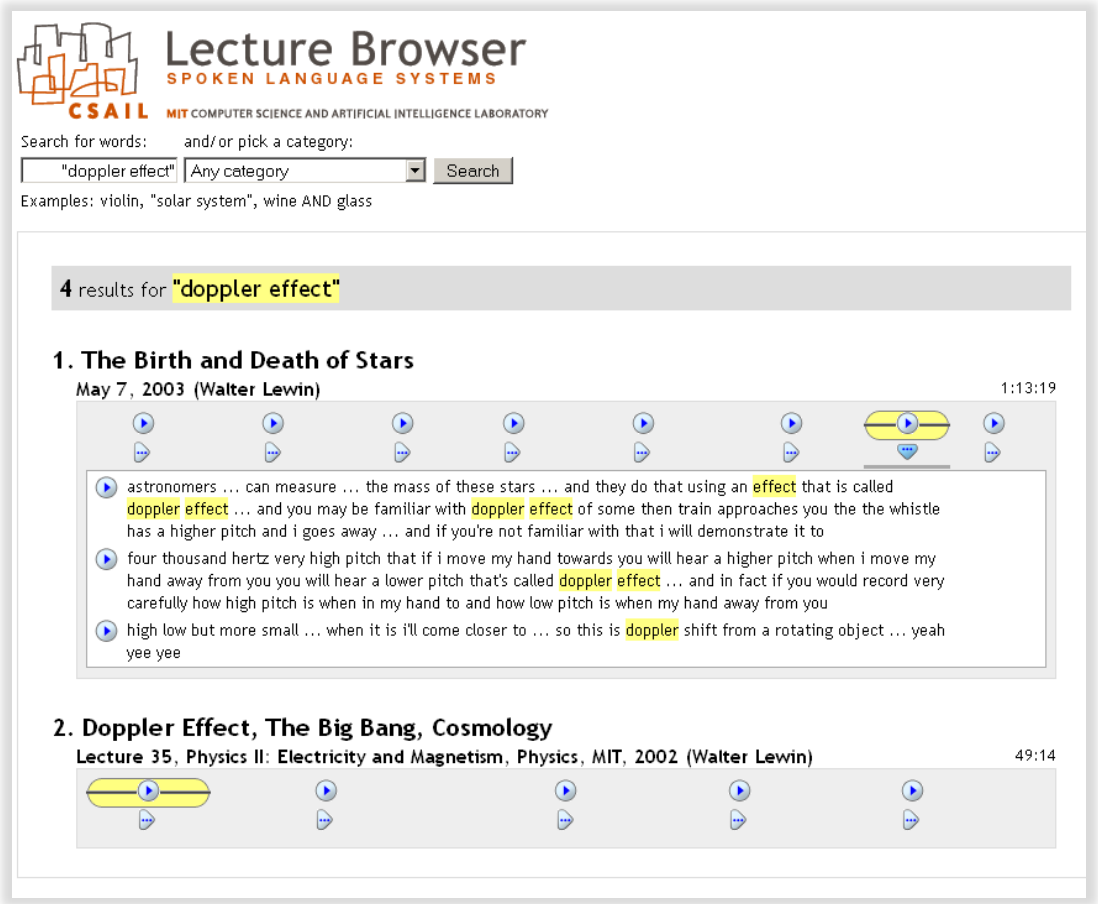


Figure 1-3: Searching within lecture transcripts in the MIT Lecture Browser system

1.3 Speech recognition accuracy

ASR systems are imperfect, and may introduce many errors into a transcription. Key factors affecting accuracy include:

1. the audio quality of the recording, influenced by the type of microphone used, venue acoustics and amount of background noise
2. whether the recognition system has been trained for a particular speaker, or is using a speaker-independent acoustic model
3. for speaker-independent systems, the match between the speaker's accent and the acoustic model
4. the match between the vocabulary and pronunciation in the lecture with the language model and pronunciation dictionary.

In an automated lecture capture system in a university, wide variations in all of the above factors are likely: lectures take place in a range of venues with different equipment and acoustics, many different lecturers are involved typically from diverse backgrounds and thus having a wide range of accents, and lectures span a range of disciplines and topics.

One can therefore expect a correspondingly wide range in the transcription accuracy produced by a large-vocabulary, speaker-independent continuous speech recognition system in this context.

Many transcripts are thus likely to fall short of the accuracy threshold for readability [7]. These transcripts are therefore unusable as a substitute for the recording itself (i.e. for a student to read as an alternative to playing back the recording), but may still be useful for search and navigation.

This project focuses on the application of ASR technologies for generating lecture transcripts for the primary purpose of information retrieval, i.e.:

- identifying lectures in which search terms occur, and
- identifying the points within a lecture where search terms occur.

The most important optimizations of an ASR system in this context are therefore those which improve the usefulness of the system for a user searching by keyword or phrase: the "searchability" of the transcript, understood as the extent to which the transcript facilitates discovery and navigation.

For search purposes, not all words in a transcript are of equal value. For example, a transcript which is accurate with respect to the terms most frequently used in searches may be more useful than a transcript with a higher overall accuracy, but lower accuracy for the designated terms. A more nuanced approach to accuracy is therefore required when the goal is to optimize searchability.

1.4 Improving searchability through adapting vocabulary and language models

This project focuses on the fourth factor affecting accuracy identified above, i.e. the linguistic match between the lecture and the ASR system's language resources.

While a typical one-hour lecture contains relatively few unique words (in the order of 1000), it is likely to include an abundance of unusual words reflecting the specialist vocabulary of the discipline. This may be seen in examining the distribution of words used in a lecture by word frequency rank, i.e. the position of the word in a list of English words ranked in descending order of their frequency of use in general English.

Figure 1-4 shows a comparison of word frequency by dictionary rank for a lecture on a specialist topic (Chemical Pathology of the Liver [8]) compared to a fictional text (Alice's Adventures in Wonderland [9]). While frequency-rank plots of texts in general exhibit a Zipfian distribution (an inverse power law), the lecture text in this example contains many more outliers than the fictional text.

These are shown circled, indicating words with document frequency greater than 3, and dictionary rank from approximately 25,000 to 1,000,000. For example in the Chemical Pathology lecture, "transaminases" occurs 16 times with a word frequency ranking of 143,006.

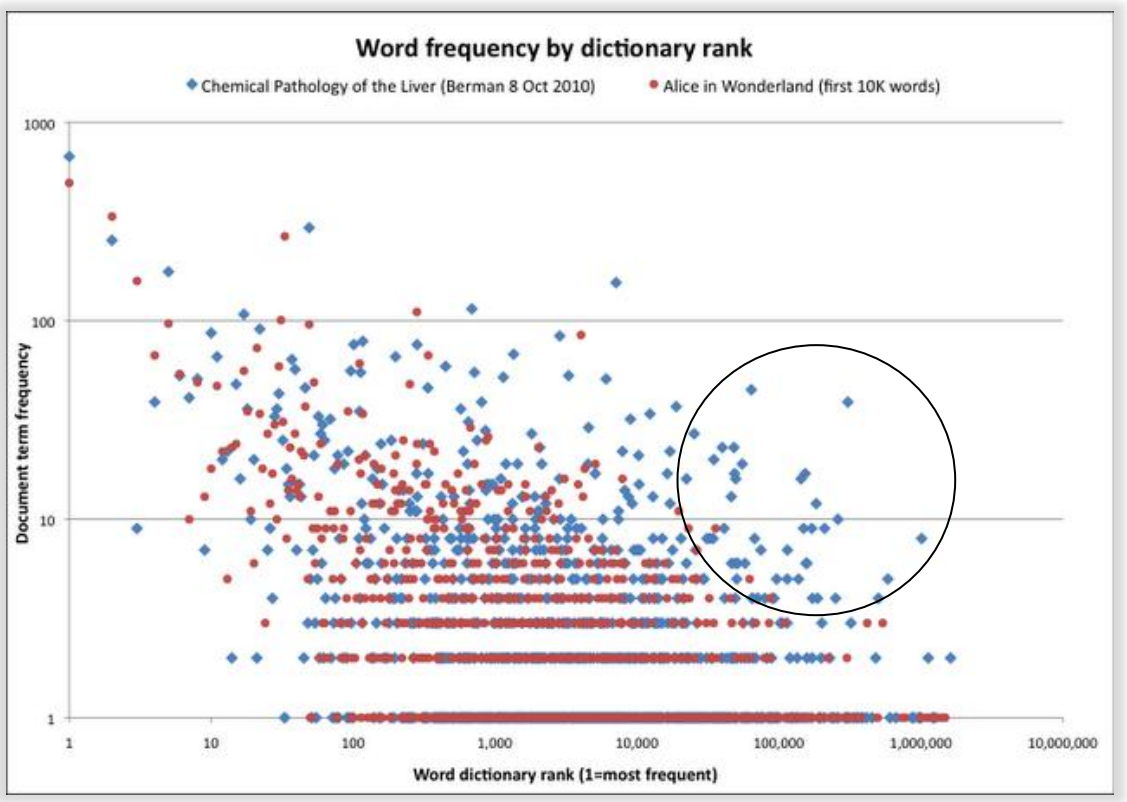


Figure 1-4: Comparison of word frequency by rank in a lecture (blue) and fictional text (red)

These outlier words are disproportionately important, as they are likely to be topic words, reflecting the content of the lecture. For search purposes, it is therefore important that ASR engines correctly recognize the unusual words, as they are likely to be strongly represented in keyword searches.

Furthermore, for most ASR engines, vocabulary represents a "hard" constraint: while other factors such as audio noise or accent mismatch may be present to a greater or lesser degree and influence the accuracy accordingly, if a word is not contained in the dictionary and language model, it will never be recognized.

While intuitively it may seem desirable to use a very large dictionary for speech recognition, the sample lecture above would require a dictionary of more than a million words to encompass more than 99% of the unique words used [10]. This would in turn require a correspondingly large statistical language model.

Unfortunately, large generic language models present significant performance and accuracy challenges for the current generation of ASR systems. The larger the model, the greater the search space, which slows down recognition and degrades accuracy given that there are many more hypotheses for each word. More importantly, such models lose the essential advantage of context; for example that a lecture on the Chemical Pathology of the Liver is unlikely to include a discussion of existential philosophy.

A desirable goal therefore is a language model and dictionary specific to the topic of the lecture (highly attuned to the context), allowing the language model to be small enough

to achieve good performance and accuracy, while being optimized for the recognition of terms in the lecture most likely to be used for search and navigation.

This project therefore investigates the unsupervised adaptation of language models to the topic of a lecture, with the assumptions that:

- lectures may cover a wide range of topics and disciplines
- in the context of a largely automated enterprise lecture capture system, little would be known about the content of the lecture in advance, other than the course name and lecture title.

1.5 Using Wikipedia as a linguistic resource for language model adaptation

Text corpora used for language modelling are often curated from within a specific genre (for example the HUB4 corpus, derived from broadcast news [11]). By contrast, the loosely-curated English Wikipedia is an attractive linguistic resource for this application because it is extremely large (containing millions of articles), constantly evolving, wide-ranging in content, and semantically dense through an abundance of inter-article links.

A subset of Wikipedia articles is identified which relate to the topic of the lecture. The text from those articles is then used as language corpus to create a topic-specific language model. As the topic of a lecture is not a well-defined concept (nor is a definitive mapping possible from topic to vocabulary), two fuzzy measures are adopted.

Firstly, a small set of keywords is identified from the course title and lecture title, and used as search terms for the Wikipedia search service to locate a set of seed articles. Next, two alternate methods are explored to identify a set of articles to harvest, starting from the seed articles.

The first method uses a naïve web-crawler strategy to follow all links in a Wikipedia article recursively until a certain quantity of text has been harvested. The second method employs latent semantic indexing and vector space modelling to generate a similarity metric between any two given Wikipedia pages, as a proxy for article relatedness and topic relevance. The crawler strategy is then adapted to follow only links to articles which are sufficiently similar to the parent article.

1.6 Implementation and evaluation

A prototype Wikipedia article crawler is implemented to harvest text from a set of Wikipedia articles, using the strategies described above.

The CMU Sphinx4 ASR engine is then used to generate three ASR transcripts for each of thirteen recorded lectures from Open Yale Courses, using the following language models:

- the open source HUB4 language model, as a reference
- a language model adapted for each lecture using Wikipedia articles harvested using the naïve crawler strategy
- a language model adapted for each lecture using Wikipedia articles harvested using the article similarity crawler strategy.

Three standard metrics – Perplexity, Word Error Rate, and Word Correct Rate – are used to evaluate the extent to which the adapted language models improve the searchability of the resulting transcripts, and in particular improve the recognition of specialist words.

Ranked Word Correct Rate is proposed as a new metric better aligned with the goals of improving transcript searchability and specialist word recognition.

1.7 Open source

A secondary goal of the project is to demonstrate how an ASR system with dynamic topic adaptation could be incorporated into an open source lecture capture framework.

Therefore software toolkits, language resources and data sets have been selected which have appropriate open source or open content licenses.

1.8 Research questions

The main research question is:

> How can English Wikipedia be used as a language corpus for the unsupervised topic adaptation of language models to improve the searchability of lecture transcripts generated by an automatic speech recognition engine?

Sub-questions are:

- To what extent do topic-adapted language models created from Wikipedia produce more searchable transcripts than those created using a generic reference language model?

- To what extent do topic-adapted language models created from Wikipedia using a crawler strategy bound by an article similarity metric produce more searchable transcripts than those created from Wikipedia using a naïve crawler strategy?

2 Background

2.1 The field of speech recognition

Automatic speech recognition (ASR) is a broad field encompassing technologies used for multiple applications and problem domains.

Rabiner and Juang's detailed account of the fundamentals of speech recognition [12] dates the first research in the field to the early 1950s, when researchers at Bell Labs built a system to recognize single digits spoken by a single speaker. Since then, the field has drawn on the disciplines of signal processing, acoustics, pattern recognition, communication and information theory, linguistics, physiology, computer science and psychology.

Three approaches to speech recognition have been explored:

- 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.

- 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.

- 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.

Of these approaches, the statistical pattern-recognition approach produced significantly better accuracy than the acoustic-phonetic approach, and is now the dominant paradigm for speech recognition, augmented by various AI approaches. A key element in pattern recognition is the use of Hidden Markov Models (HMMs), which enables recognizers to use a statistical model of a pattern rather than a fixed template.

ASR systems are known to perform best on audio recorded using a close-talking microphone in a noise-free environment, transmitted through a clear channel, and recorded with a high sampling frequency (16 KHz or greater). However, as these conditions are seldom available in real-life, a range of strategies have been investigated to compensate for the effects of noise, reverberation and variation in conditions between the reference recordings used for training recognizers and actual recordings. While acoustic issues are not explored in depth here, they remain a significant constraint on recognition performance [13].

2.2 Core concepts in speech recognition

2.2.1 Recognition scope

Speech recognition applications can be broadly characterised in three ways: speaker dependent or independent, small or large vocabulary, and isolated or connected recognition.

Speaker-dependent systems are designed to recognize speech from one person, and typically involve a training exercise where the speaker records sample sentences to enable the recognizer to adapt to the speaker's voice. Speaker-independent systems are designed to recognize speech from a wide range of people without prior interaction between the speakers and the recognition system.

Small vocabulary systems are those where only a small set of words is required to be recognized (for example fewer than 100), and permissible word sequences may be constrained through a prescriptive grammar. Large vocabulary systems are those designed to recognize the wide range of words encountered in natural speech (for example up to 60,000 words).

Finally, isolated recognition systems are intended to recognize a discrete word or phrase, typically as an action prompt in an interaction between person and system, whereas connected recognition systems are intended to recognize continuous words and sentences following each other without interruption.

Three possible applications and their characteristics are shown in Table 2-1.

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

Table 2-1: Characteristics of some common speech recognition applications

The subfield relevant to the creation of automatic transcripts from lecture speech is thus characterised as speaker-independent (SI) large vocabulary connected (or continuous) speech recognition (LVCSR).

2.2.2 Acoustic and language models

Recognition systems following the dominant statistical pattern-recognition paradigm make use of four related resources for a given language and speaker population:

1. A set of phonemes
2. A phonetic dictionary
3. An acoustic model
4. A language model

A phoneme is a unit of sound making up an utterance. The most general representation of phonemes is that provided by the International Phonetic Alphabet (IPA), which includes orthography for phonemes found in all oral languages [14].

However, for speech recognition applications, ASCII representations of phonemes are more practical. A widely used ASCII set is the Arpabet (Table 2-2), created by the Advanced Research Projects Agency (ARPA) to represent sounds in General American English [15]. The Arpabet comprises 39 phonemes each represented by one or two letters with optional stress markers represented by 0, 1 or 2.

Arpabet symbol	Sound type	Arpabet symbol	Sound type
AA	vowel	L	Liquid
AE	vowel	M	Nasal
AH	vowel	N	Nasal
AO	vowel	NG	Nasal
AW	vowel	OW	Vowel
AY	vowel	OY	vowel
B	stop	P	stop
CH	affricate	R	liquid
D	stop	S	fricative
DH	fricative	SH	fricative
EH	vowel	T	stop
ER	vowel	TH	fricative
EY	vowel	UH	vowel
F	fricative	UW	vowel
G	stop	V	fricative
HH	aspirate	W	semivowel
IH	vowel	Y	semivowel
IY	vowel	Z	fricative
JH	affricate	ZH	fricative
K	stop		

Table 2-2: The Arpabet phoneset without stress markers from the CMU Pronouncing Dictionary

The relationship between words and phonemes is captured in a phonetic dictionary (or pronouncing dictionary), which maps each word to one or more sets of phonemes. Table 2-3 shows examples with stress markers for *aardvark, tomato* and *Zurich* (including an alternate pronunciation for *tomato*) from the CMU Pronouncing Dictionary 0.7a [16].

Word	Arpabet Pronunciation
AARDVARK	AA1 R D V AA2 R K
TOMATO	T AH0 M EY1 T OW2
TOMATO(1)	T AH0 M AA1 T OW2
ZURICH	Z UH1 R IH0 K

Table 2-3: Examples of Arpabet pronunciations from the CMU Pronouncing Dictionary

Pronouncing dictionaries are used both for speech-to-text applications (speech recognition), and text-to-speech applications (speech synthesis).

An acoustic model associates features from the sound signal with phonemes. As the pronunciation of an individual phoneme is affected by co-articulation effects (how sounds are pronounced differently when voiced together), many systems model phoneme triples, i.e. a phoneme in context of the phonemes preceding and following it. As the exact pronunciation and sound of a phoneme may vary widely, even from a single speaker, acoustic models reflect probabilities that a set of acoustic features may represent a particular phoneme (or set of phonemes).

Acoustic models are trained from a speech corpus consisting of audio recordings matched with a transcription. The transcription typically contains time-alignment information to the word- or phoneme level. Speaker-independent models are trained with audio from a wide range of speakers (for example with a mix of male and female speakers and regional accents). Speaker dependent models may be trained from a single speaker, or more commonly, created by adapting a speaker independent model to a given speaker.

However, acoustic models alone are insufficient to achieve acceptable levels of accuracy, as can be illustrated by the challenges of disambiguating between homonyms and similar-sounding phrases such as “wreck a nice beach” and “recognize speech”. Linguistic context is thus an additional and indispensable resource in generating plausible recognition hypotheses.

The dominant approach to language modelling is the n-gram language model (LM). Such language models are trained from a text corpus, and give the probability that a given word will appear in a text following the (n-1) preceding words. Smoothing techniques are often applied to the initial model to adjust the probabilities to compensate for the fact that less frequent words which have not been seen in the training text may also occur.

For example, Table 2-4 shows the probabilities for words which might follow “your economic” in a trigram (3-word) language model in ARPA format:

Log10 Probability	Trigram
-2.0429	YOUR ECONOMIC ADVISERS
-1.2870	YOUR ECONOMIC FUTURE
-2.0429	YOUR ECONOMIC GROWTH
-1.7585	YOUR ECONOMIC POLICIES
-1.7585	YOUR ECONOMIC POLICY
-1.1613	YOUR ECONOMIC PROGRAM
-2.0429	YOUR ECONOMIC PROGRAMS
-1.5947	YOUR ECONOMIC PROPOSALS
-2.0429	YOUR ECONOMIC REFORM
-2.0429	YOUR ECONOMIC REFORMS
-1.3695	YOUR ECONOMIC TEAM

Table 2-4: Excerpts from a Trigram Language Model trained from the HUB4 Corpus

In this example, where the recognizer is assessing which hypothesis is most likely for a word following "your economic", the language model would favour "program" rather than "programs", and "team" over the homonym "teem". However, the model would give no advantage to the recognizer in distinguishing between singular and plural forms of "reform" and "policy" as they are equally likely in the model.

Language models are used in a range of natural language processing applications, including spell-checkers (to suggest the most likely correction for a misspelt word) and machine translation systems (translating words, phrases and sentences from one language to another).

2.2.3 Speech corpora

Training acoustic and language models require appropriate corpora. Notable corpora used in speech recognition research have included:

- The TIMIT corpus of American English speech (1986), which consists of a set of sentences each read by a range of different speakers [15].
- The Wall Street Journal (WSJ) corpus (1992), derived largely from text from the Wall Street Journal newspaper from 1987-1989 read aloud by a number of speakers [17].
- The HUB4 English Broadcast News Speech corpus (1996/7), generated from transcriptions of news programmes broadcast in the United States on CNN, CSPAN and NPR [11], [18].
- The Translanguage English Database (TED) corpus (2002), created from lectures given by a range of speakers at the Eurospeech '93 conference [19].

The above examples have each been carefully curated to serve research purposes, and are derived from specific genres or application domains. Models trained from such corpora may be less effective when applied to different contexts. For example, acoustic models trained by American English speakers may be less effective for recognizing

speech from other parts of the world, and language models trained on broadcast news may be less effective when applied to a different genre, such as poetry.

2.2.4 Supervised and unsupervised model adaptation

To improve the alignment between acoustic and/or language models and the speaker and genre of text being recognized, it can be more effective to adapt an existing model with a limited amount of new training data, rather than create an entirely new model. This is especially the case if the volume of new training data is insufficient to create a robust model from scratch.

Supervised adaptation refers to the process of adapting models with some manual intervention based on prior knowledge of the target speaker and domain. Examples include adapting an acoustic model with transcribed sentences from the speaker, or adapting a language model with material from a textbook related to the topic being spoken about. Supervised adaptation may produce good results, but limit the generality of the approach.

Unsupervised adaptation is when the recognition system adapts the acoustic and/or language models in response to the input data provided for the recognition task. Such adaptation is often performed iteratively, with output data from a first-pass recognition attempt used to modify the models used for subsequent recognition passes.

Further examples are given in 2.5 and 2.6.

2.3 Applying speech recognition systems to lectures

Over the last decade, a number of research groups and projects have undertaken systematic work in applying speech recognition to lectures, progressively investigating multiple techniques and approaches. Major programmes include:

- work by the Spoken Language Systems Group in the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT [5]
- work by Cosmin Munteanu and colleagues in the Computer Science Department at the University of Toronto [20]
- the Science and Technology Agency Priority Program in Japan, “Spontaneous Speech: Corpus and Processing Technology”, supporting work particularly at Kyoto University and the Tokyo Institute of Technology [21]
- the Liberated Learning Project [22]
- the Net4voice project under the EU Lifelong Learning Programme [23]
- the Computers In the Human Interaction Loop (CHIL) project under the EU FP6 [24].

The starting point of speech recognition research for lectures is usually recognition systems developed for earlier applications. These include broadcast news and meeting transcription systems, or speaker-dependent systems such as those used for dictation. Speaker independent systems are typically trained with widely available speech and language corpora, such as those described in 2.2.3.

As initial results in applying the recognition systems and accompanying acoustic and language models to lecture speech usually produced poor results characterised by high error rates, much of the related research effort has focused on improving the effectiveness of speech recognition for lectures through different types of generalization and specialization of earlier systems and approaches.

Generalization approaches have examined ways of accounting for the larger vocabulary, including specialized terms, and greater variation in delivery style characteristic of spoken lectures. Specialization approaches have looked at features specific to many lectures, such as the use of presentation slides, and using these attributes to "know more" about the content of the lecture and thus improve recognition accuracy and usefulness.

A further class of research starts by accepting the imperfect nature of automatically generated transcripts, and examines how to involve users in improving transcript accuracy and where possible use correction feedback to further improve subsequent automated recognition tasks.

2.4 Modelling the form, style and content of lectures

The form and linguistic style of lectures present both challenges and opportunities for ASR systems.

For example, Yamazaki et al note the high level of spontaneity in lectures, which are characterized by "strong coarticulation effects, non-grammatical constructions, hesitations, repetitions, and filled pauses" [25]. Glass et al note a colloquial style dramatically different to that in textbooks, characterized by poor planning at the sentence level and higher structural levels [26]. Lectures additionally exhibit a high number of content-specific words which may not occur in a general speech corpus. Spoken and written forms of language may diverge differently in different languages; for example, Akita and Kawahara note significant linguistic differences between spoken and written Japanese [27].

These variations have presented recognition difficulties, and a range of strategies have been explored to compensate.

Structural features in the genre have been observed and exploited to improve recognition performance. Such features include rhetorical markers, the use of presentation slides, a close correspondence between speech and slides or textbook, and affinity between the content of related lectures and between lectures and associated material available on the web.

Most models of the lecture for ASR systems assume a single speaker engaged in a monologue in a single language, accounting for students or the audience only so far as they constitute a potential source of noise. Birnholtz notes a lack of "systematic study of face-to-face behavior" in the research related to webcasting systems, focusing particularly on audience interactivity and how turn-taking ("changes in floor control") is dynamically negotiated [28].

Although a sub-field of speech recognition known as speaker diarization is devoted to identifying multiple speakers in audio (typically in the context of meetings or conferences) [29], the potential requirement for ASR systems to transcribe not only the speech of the lecturer but also that of people asking questions or interjecting in a lecture is largely unexplored.

2.5 Acoustic model adaptation

Acoustic models derived from the broadcast and news genres may be a poor fit for lecture recordings, and thus a class of research has focused on how to adapt acoustic models to more accurately reflect the characteristics of lecture speech.

Adaptation strategies which have shown some success include accounting for non-linguistic speech phenomena ("filler" sounds) [30], dynamically adjusting the model to account for speaking rate [31], unsupervised adaptation to account for new speakers [32] and using discriminatively trained models for language identification and multilingual speech recognition [33].

2.6 Language model adaptation

Researchers have investigated strategies for generating and adapting the language model (LM) to improve recognition accuracy for lectures, on the assumption that a model which closely reflects the context of the utterances is likely to outperform a more generic language model. Adaptations have been investigated for three levels of context:

- at the macro level, for all lectures, treating spoken lectures as a genre with distinct characteristics
- at the meso level, for a single lecture, taking advantage of prior knowledge about the lecture topic or speaker
- at the micro level, for a part of a lecture, using knowledge about segments or transitions within a lecture.

Many adaptation strategies make use of some prior knowledge or parallel media. This could include information about the topic or knowledge domain of the lecture, a textbook or instructional materials related to the course or the lecture presentation slides. Use of such information may provide specific improvements at the expense of the generality of the technique (for example, not all lectures may be accompanied by slides).

Kato et al investigated the use of a topic-independent LM, created by creating a large corpus of text from lecture transcripts and panel discussions, with topic-specific keywords removed [34]. The model is then adapted to specific lectures by using the preprint paper of the lecture to be delivered (when available).

Willett et al propose two iterative methods of unsupervised adaptation [35] [36]. Both methods show improvements in accuracy up to the second iteration of application.

A first method identifies texts from a large corpus which are considered close to the first-pass recognition text by using a Term Frequency – Inverse Document Frequency (TF-IDF) measure, and uses the selected texts to adapt the LM. TF-IDF is a weighting factor which assigns a score to the importance of the word based on its occurrence in

the document (term frequency) but adjusted to avoid words which are common across all documents (such as “a” and “the”) from dominating the score.

A second method uses a minimum discriminant estimation (MDE) algorithm to adapt the LM, following the thesis that “seeing a word uttered at some place within the speech increases the likelihood of an additional appearance”. MDE is a technique for adapting a language model to more closely match the distribution of words seen in the recognized text, while minimizing the variation from original to adapted model, using a measure of distortion (or discrimination information) known as the Kullback- Leibler distance. [37]

Nanjo and Kawahara report similar work, and further explore adaptations to the lexicon and LM to account for variant pronunciations [38].

The use of lecture slides for adapting the LM has been explored by several research groups. Yamazaki et al note that a “a strong correlation can be observed between slides and speech” and explore first adapting the LM with all text found in the slides, then dynamically adapting the LM for the speech corresponding to a particular slide [25].

Munteanu et al pursue an unsupervised approach using keywords found in slides as query terms for a web search. The documents found in the search are then used to adapt the LM [39].

Kawahara et al investigate three approaches to adapting the LM, viz. global topic adaptation using Probabilistic Latent Semantic Analysis (PLSA), adaptation with web text derived from keyword queries and dynamic local slide-by-slide adaptation using a contextual cache model. They conclude that the PLSA and cache models are robust and effective, and give better accuracy than web text collection because of a better orientation to topic words [40]. Latent Semantic Analysis is an approach to document comparison and retrieval which relies on a numeric analysis of word frequency and proximity.

Akita and Kawahara propose a statistical transformation model for adapting a pronunciation model and LM from a text corpus primarily reflecting written language to one more suited for recognizing spoken language [27].

While n-gram language models are the dominant paradigm in ASR systems, they offer a relatively coarse model of language context. Newer research is exploring more accurate statistical representations of “deep context”, for example accounting for connections between related but widely separated words and phrases [41].

2.7 Measuring the accuracy of lecture transcripts

The most widely used accuracy metric for recognition tasks is the Word Error Rate (WER), computed as the Levenshtein distance (“edit distance”) between the recognized text and a reference transcript. This is the number of insertions, deletions and substitutions required for the hypothesis to match the reference transcript, as a proportion of the number of words in the reference transcript. A related measure is the Word Correct Rate (WCR), which ignores insertion errors.

WER is often used as a measure of readability and thus comprehension task performance. Munteanu investigated the usefulness of transcripts with a range of different error rates, showing that transcripts with a WER of 25% were perceived to be as useful as manually-generated transcripts. When examining user scores on a quiz testing information recall after viewing a video with transcript, a linear relationship emerged between WER and quiz performance. At a lower bound of 45%, quiz performance was worse than having no transcript at all. However, the study reports that users' perception of transcript quality is subjective, coarse-grained and task-dependent [20].

While its widespread use makes WER a useful measure to compare competing approaches, it may often not account for the actual impact of errors for the application at hand. For example some errors may be more trivial than others and easily overlooked, while keyword accuracy may be disproportionately significant.

Bourlard et al have taken issue with WER's dominance in the field, arguing that reliance on reporting WER may in fact be counter-productive, undermining the development of innovative new approaches and "deviant" research paradigms [42].

McCowan et al point out that WER characterises recognition performance as a string editing task, whereas for many applications speech recognition is better understood as supporting information retrieval tasks [43]. Cited weaknesses of WER include that it is not a proper rate (as it can range below 0 and above 1), is not easily interpretable and cannot be decomposed in a modular way.

Park et al examine automatic transcripts from the perspective of information retrieval (IR), investigating the effects of different recognition adaptations on WER and the IR measures precision and recall, which relate to matches of keyword query strings in the recognized text [44]. Results show that good retrieval performance is possible even with high error rates, and conversely that adapting the language model with spontaneous speech data improves accuracy, but is of marginal value to information retrieval tasks. Wang et al argue against WER, reporting results where an alternate language model produced higher word error rates but better performance with an alternate task-oriented metric, slot understanding error [45].

McCowan et al propose four qualities for an improved metric: that it should be a direct measure of ASR performance, calculated in an objective, automated manner, clearly interpretable in relation to application performance and usability, and modular to allow application-dependent analysis [43].

2.8 Prototype implementations of speech recognition for recorded lectures

A number of prototype applications have integrated ASR systems with lecture recording and playback systems, enabling the end-user to interact with the transcript text. These use the time-alignment information generated by the ASR process to synchronize the transcript with the video playback. Words or phrases in the transcript text act as index markers into the recording, allowing the user to click on a point in the transcript to play

back the audio and/or video from the corresponding point. A further common feature is text search, which highlights matching points in the transcript or on a timeline.

2.8.1 ePresence

At the University of Toronto, Cosmin Munteanu and colleagues extended the ePresence system [46] with transcripts generated by the SONIC ASR toolkit [47], [48].

The project is comprehensively described in Munteanu's PhD thesis [20], as well as in a number of separate papers exploring accuracy rates [7], web-based language modelling [39], collaborative editing for improving transcripts [49], and the application of transformation-based rules for improving accuracy with minimal training data [50].

The integration of the transcript in the user interface is shown in Figure 2-1 [51].

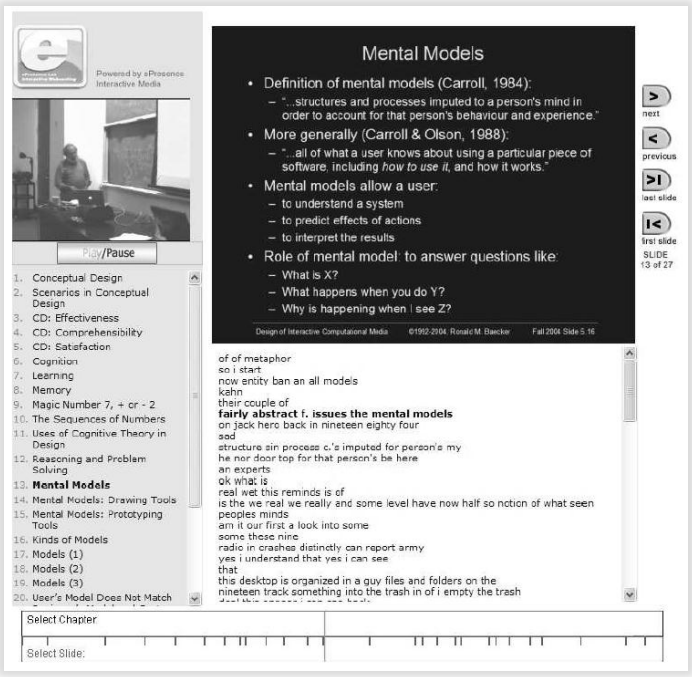


Figure 2-1: Prototype of ASR extensions to the ePresence system

2.8.2 MIT

The Spoken Language Systems Group in the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL) has investigated multiple dimensions of speech recognition in a long-running research programme. The group's SUMMIT recognizer [52], [53] has been applied to lecture recordings to produce the Lecture Browser, shown in Figure 2-2 [54].

Publications from the group have reported inter alia on experimental results with lecture recordings [55] and explored the linguistic characteristics of lectures [26], vocabulary selection and language modelling for information retrieval [44], pattern discovery in lectures [56], approaches to error correction [57], pronunciation learning from continuous speech [58] and approaches to crowdsourcing the generation of language models and transcripts [59] [60].

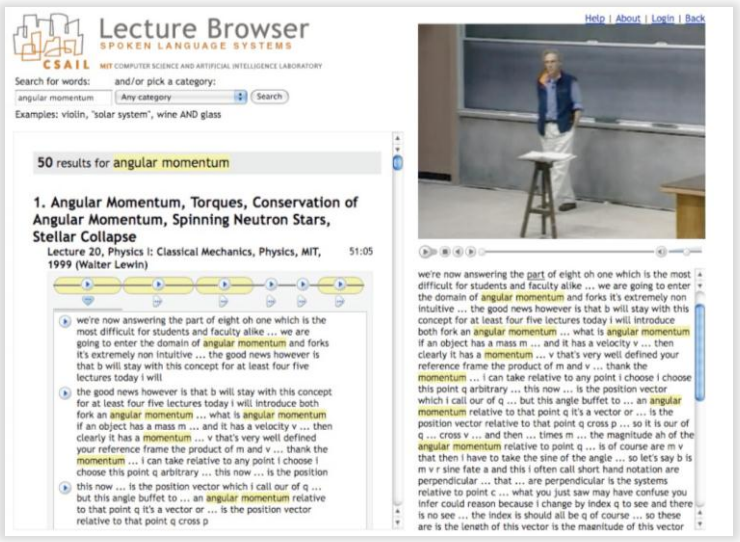


Figure 2-2: The MIT Lecture Browser

2.8.3 Synote

Wald and colleagues at the University of Southampton and the Liberated Learning Consortium have focused on using automated transcripts to make lectures more accessible for deaf and hard of hearing students [61]. The Synote application uses a recognizer developed by IBM and the LL Consortium, ViaScribe, presenting a web user interface which also allows bookmarks and annotations, shown in Figure 2-3 [55].

The project has focused on real-time display in teaching venues, while noting that as ASR engines may be configured to optimize for accuracy over speed, the same lecture could be re-processed afterwards to create a more accurate transcript for online use.

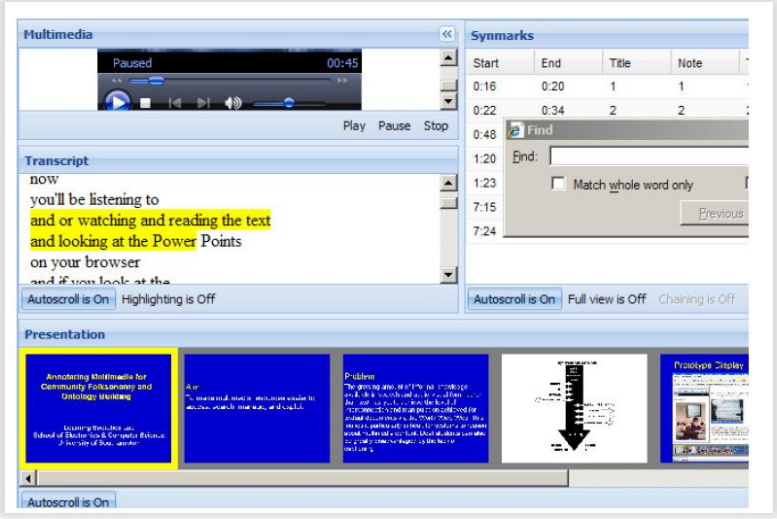


Figure 2-3: The Synote Annotation System

2.8.4 REPLAY

REPLAY is an open source lecture capture system developed at ETH Zürich [62].

A prototype for including ASR transcripts in REPLAY generated by the CMU Sphinx 4 recognition engine [63] was developed and described by Samir Atitallah [64], shown in Figure 2-4.

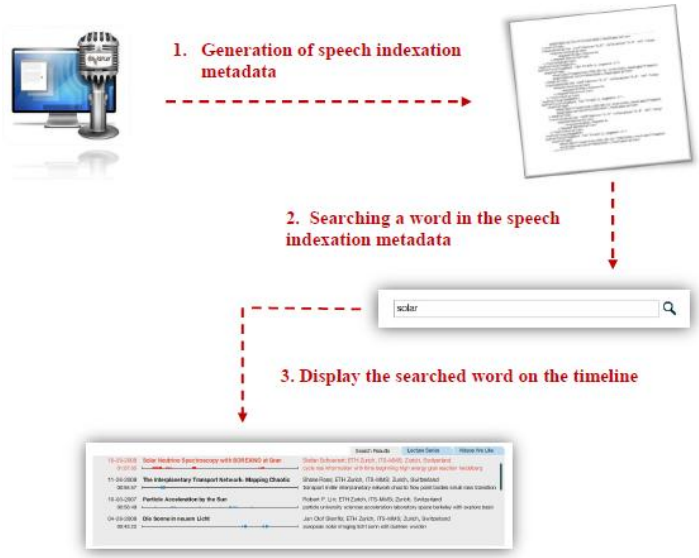


Figure 2-4: Speech recognition in REPLAY

The project examined software implementation strategies, appropriate metadata formats for storing timestamped transcripts and audio- and word-based optimization strategies for improving accuracy.

2.9 Alternate approaches and extensions

2.9.1 Real-time transcription

A decade of research in the Liberated Learning Project has highlighted the value of real-time transcription and captioning, using commercially available recognition engines [22] [61] [65] [66]. Whereas many systems with integrated transcripts are designed for post-lecture viewing and recall, real-time systems support the participation of deaf or hard-of-hearing students during the lecture itself.

2.9.2 Improving transcripts with user input

Despite the many incremental improvements in recognition performance described above, ASR systems are still not regarded as being robust or accurate enough to always produce usable and useful transcripts.

Researchers have therefore turned to user input as a strategy to close the “error gap”, which is in the range of 10-25%. Munteanu first explored whether students would be prepared to invest effort in correcting transcripts online, finding positive results with a range of incentives: in a field trial using a wiki-like online editor, 84% of all transcript lines were edited by users [20].

2.9.3 Indexing, segmentation and searching

Many ASR systems aim to generate a complete transcript of a lecture and have the goal of optimizing the accuracy of the transcript. However, a transcript is often just the means to an end. A range of alternate approaches have been proposed which pursue the goals of indexing, keyword extraction, segmentation or search directly, without requiring an accurate or complete transcript.

Yamamoto et al observe that a lecture may consist of several topics closely aligned with a textbook, and demonstrate a method of topic segmentation by comparing topic vectors from speech and textbook, constructed using nouns weighted using a TF-IDF measure.

Lin et al investigate segmentation using multiple linguistic features [67]. Five content-based features and two discourse-based features are used to create feature vectors, which are used to compare the similarity of adjacent sections of text.

Kawahara et al approach segmentation using presumed discourse markers, "expressions that are characteristic to the beginning of new sections in lectures and oral presentations", describing an unsupervised training approach for the markers [68].

Seide et al describe a non-textual approach to search. Rather than matching search text against a transcription, the recognizer "generates lattices of phonetic word fragments, against which keywords are matched phonetically" [69]. The vocabulary- and domain-independent approach is shown to be as accurate as vocabulary/domain-dependent systems, and has the advantage of maintaining this accuracy for out-of-vocabulary (OOV) words.

Ngo et al describe a segmentation approach for video where the camera field of view captures both the speaker and projected slides [70]. The video is segmented by identifying the slide transitions, recognizing text on projected slides, and extracting phrases and keywords to constrain speech recognition to identify only the sought-after words for the purpose of aligning topics with video segments.

Repp and Meinel describe techniques for semantic indexing of lectures [71]. A generated thesaurus is used to associate common phrases with pedagogically meaningful "meta-phrases", for which the authors suggest "example", "explanation", "overview", "repetition" and "exercise". Identified meta-phrases in the recognized text are then used as index keys to the video, allowing students to navigate according to their learning objective.

Repp et al further explore another indexing technique, using word repetitions as an indicator of thematic cohesion [72]. "Chain indexes" are created for an audio search tool, which presents results as matching segments of the video timeline.

Park and Glass apply approaches suggested by developmental psychology and comparative genomics to identify recurring speech patterns directly in the acoustic signal [56]. The patterns are grouped into clusters which correspond to lexical entities such as words and short phrases. The identified clusters are shown to have high relevance to the content of the lecture.

Liu et al explore unsupervised approaches to keyword extraction from transcripts [73]. TF-IDF weighting, part-of-speech, word clustering and sentence salience scores are shown to be of value using a variety of evaluation methods.

Many of the surveyed techniques are designed to exploit particular features of the media (for example linguistic attributes of a lecture) and thus attain improved results at the expense of the generality of the technique. On the other hand techniques which appear more generalizable introduce other types of constraints. For example, audio search algorithms require a specialized search service, which could limit the visibility of published media to text-based search engines on the open web.

Thus while alternate approaches for indexing, segmentation and searching appear promising, they present partial solutions which do not yet provide the full range of affordances of a complete text transcript.

2.9.4 Improving manually-generated transcripts with ASR

Hazen outlines an inverted approach, where it may be feasible to obtain imperfect human-generated transcriptions quickly or cheaply [74]. Speech recognition technologies may then be used for automatic alignment of the text with the speech, to discover and automatically correct transcription errors.

The results show an improvement in word error rate for the adjusted transcript over that produced by a human transcriber, but also show that most of the corrections represent re-insertion of omitted words which are mostly not of significance for comprehension of the text.

2.10 Remaining problems and future directions

With respect to speech recognition in general, Baker et al suggest six "grand challenges" for the field [13]:

- everyday audio (greater robustness in adverse conditions)
- rapid portability to emerging languages
- self-adaptive language capabilities
- detection of rare, key events
- cognition-derived speech and language systems
- spoken-language comprehension.

With respect to speech recognition of lectures, Munteanu suggests inter alia:

- investigating topic-specific language modelling approaches and refining ASR transcriptions for lectures and presentations that do not use slides or make use of other visual aids
- developing a user-motivated measure of transcript quality
- maximizing the trade-off between user editing and ASR improvements, and
- exploring other collaborative approaches to webcast usability improvement [20].

Tibaldi et al present a methodology “focused on exploiting and assessing the impact of Speech Recognition technology in learning and teaching processes”, an under-examined but central topic if the full benefits of ASR systems are to be realized in educational contexts [75].

2.11 Summary

A number of prototype systems have shown that ASR systems can be integrated with lecture capture systems to produce useful results. However, the central issue in the application of speech recognition to lectures is recognition performance.

Much research has focused on demonstrating small, incremental improvements to accuracy rates through innovation in algorithms or smarter adaptations to the acoustic or language models. Many posited improvements take advantage of particular features of lectures, specific to the language, domain, content, supporting media or style of presentation. Most experimental results are reported on in relation to a narrow corpus in controlled conditions.

The most widely used measure for accuracy, Word Error Rate, is not regarded as optimal for information retrieval tasks and application-specific alternate metrics are seen as more helpful in evaluating the success or failure of adaptations to language models.

Promising work has been done on examining user needs and behaviour more closely. Productive directions include harnessing human intelligence to close the “recognition gap”, and identifying the best ways to use imperfect recognition results effectively rather than seeking completely faithful transcription.

3 Methodology

3.1 Introduction

Chapters 3 and 4 describe the methodology used to investigate and evaluate the research questions posed in Chapter 1. An applied, experimental research design is used. This follows common practice in the field, and enables the impact of different language models on accuracy and searchability to be assessed across a set of real-world test cases.

In this chapter, the concept of "searchability" is characterised, leading to the identification of related metrics. A generic speech recognition process is set out, followed by details of the CMU Sphinx speech recognition engine and the selected reference acoustic and language models.

A set of recorded lectures is identified for experimentation, and the speech recognition process used with reference and custom language models is shown. Finally, the process of calculating the evaluation metrics is set out.

3.2 Aspects of searchability

Whether a lecture transcript is more or less "searchable" has multiple dimensions. Consider three scenarios:

- A member of the public would like to know more about the work of John Milton, and so does a Google Search on his name.
- A student enrolled in an English Literature course has been set an essay on a particular work by Milton, and would like to find the lecture about Milton which she recalls attending. She searches for "Milton" on the university's video portal.
- A student is playing back the recording of a 45 minute lecture, but doesn't have much time and would like to skip to the parts where Milton's friend Charles Diodati is mentioned.

In these examples, the search scope ranges from the entire Internet to a single recording. In the first two examples, the objective is discoverability: the lecture should appear in the set of search results. In the last example, the objective is better navigation within the media.

Searchability here thus encompasses discoverability (does the transcript facilitate the user finding the lecture) and usefulness (does the transcript provide fine-grained indexing).

Many factors could affect the outcome of the user's search, including the search terms chosen by the user, and the indexing, ranking and search algorithms of the search engine, which may be opaque, proprietary and evolve frequently.

The best case for maximising a lecture's searchability (given that user and search engine behaviour are both unknown to a degree) is therefore a completely accurate transcript.

However, given two imperfect transcripts of the same lecture (as will be the case for ASR-generated transcripts in the foreseeable future), which is more likely to be searchable?

For the purposes of this investigation, three assumptions are made:

- Users develop expertise in online searching, and form a mental model which leads them to prefer search terms with greater discriminatory power, to avoid being swamped with irrelevant search results. Thus users search for keywords specific to the content being sought. An alternate strategy could be searching for a distinctive sequence of words such as a short quotation.

- Thus words which occur in the document but are less frequent in general English (or the set of documents within the search scope) are disproportionately important. Thus in a transcript, "Milton" is a more valuable word for the purposes of maximising discoverability than "here".

- The introduction of extraneous words into a transcript may be harmful for the quality of search results overall, but is not significant in considering the searchability of an individual document. Thus false negatives (words incorrectly not matched) are more important than false positives (words incorrectly matched).

3.3 Selection and definition of metrics

Three primary metrics are used to assess the likely searchability of a lecture transcript. Each metric is derived from a word-by-word comparison of a reference transcript to an imperfect hypothesis transcript. Evaluation is thus automated and quantitative, and does not take into account human factors or the influence of algorithms in the selection and application of search terms.

Two metrics used which are common in speech recognition research are:

- Word Error Rate (WER). WER is an accuracy metric, calculated from the "edit distance" between two documents: the number of word insertions, deletions and substitutions required to transform the reference to hypothesis, as a proportion of the word count in the reference.

- Word Correct Rate (WCR). WCR is the number of words correctly recognized as a proportion of total word count in the reference. WCR thus ignores the effect of insertions.

To better characterise searchability in terms of keyword recognition, a new metric is introduced, Ranked Word Correct Rate (RWCR). RWCR calculates the total recognition rate of those words in the transcript which occur below a given frequency rank in general English. Thus the recognition accuracy of unusual words (e.g. "Comus") affects the recognition score, while the recognition accuracy of common words (e.g. "a", "the", "and") is ignored.

The method of calculation and examples of the above metrics are shown in 3.9.

Secondary metrics which give insight into aspects of the recognition process are:

- Vocabulary coverage, expressed by the number of out-of-vocabulary words (i.e. words found in the transcript which are not included in the recognition dictionary and language model)

- Vocabulary recognition, expressed by the number of unrecognized words (words in the transcript which are in the recognition dictionary and language model, but do not occur in the hypothesis) and extraneous words (words which do not occur in the reference, but were incorrectly introduced to the hypothesis).

- The perplexity of the language model (evaluated against a reference text), which is an information theory measure expressing the extent of the uncertainty which the recognizer might face in selecting word hypotheses.

3.4 Generic speech recognition process

Figure 3-1 illustrates a generic speech recognition process using a Hidden Markov Model (HMM) recognition engine with a statistical language model.

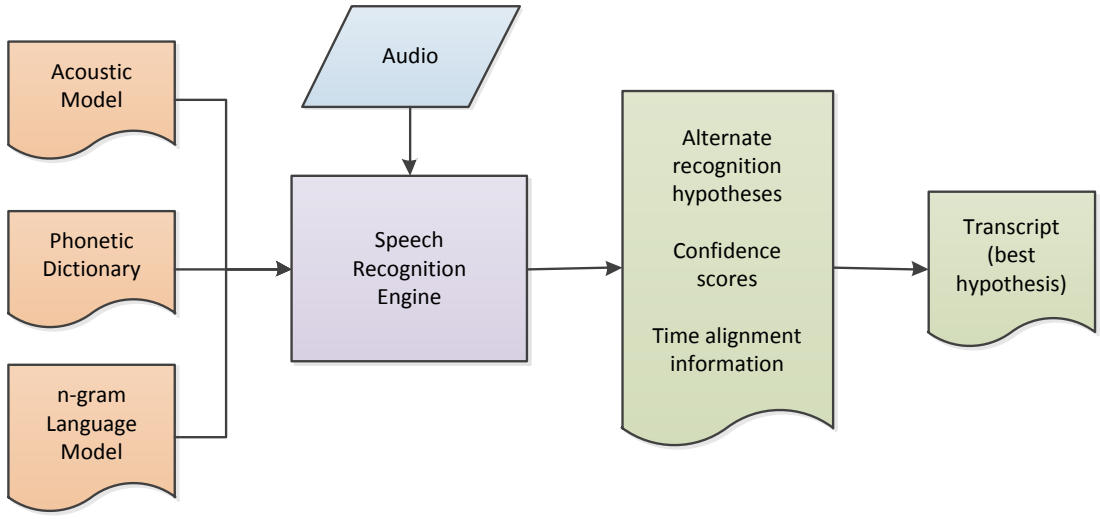


Figure 3-1: Speech recognition with a statistical HMM engine

The recognizer makes use of an acoustic model, phonetic dictionary and n-gram language model to recognize audio from an input file.

Depending on configuration, the recognizer may output several different recognition hypotheses for each word or phrase with confidence scores, or a single best hypothesis. Time alignment information may be output for use in applications such as search navigation or video subtitling.

For this investigation, only the best hypothesis plain text transcript is used.

3.5 The CMU Sphinx ASR engine

CMU Sphinx is an open source speech recognition toolkit from Carnegie Mellon University. The version of Sphinx selected for this project is Sphinx4, a highly customized recognition engine written in Java. Figure 3-2 illustrates the Sphinx4 architecture, as described in the Sphinx4 White Paper [63].

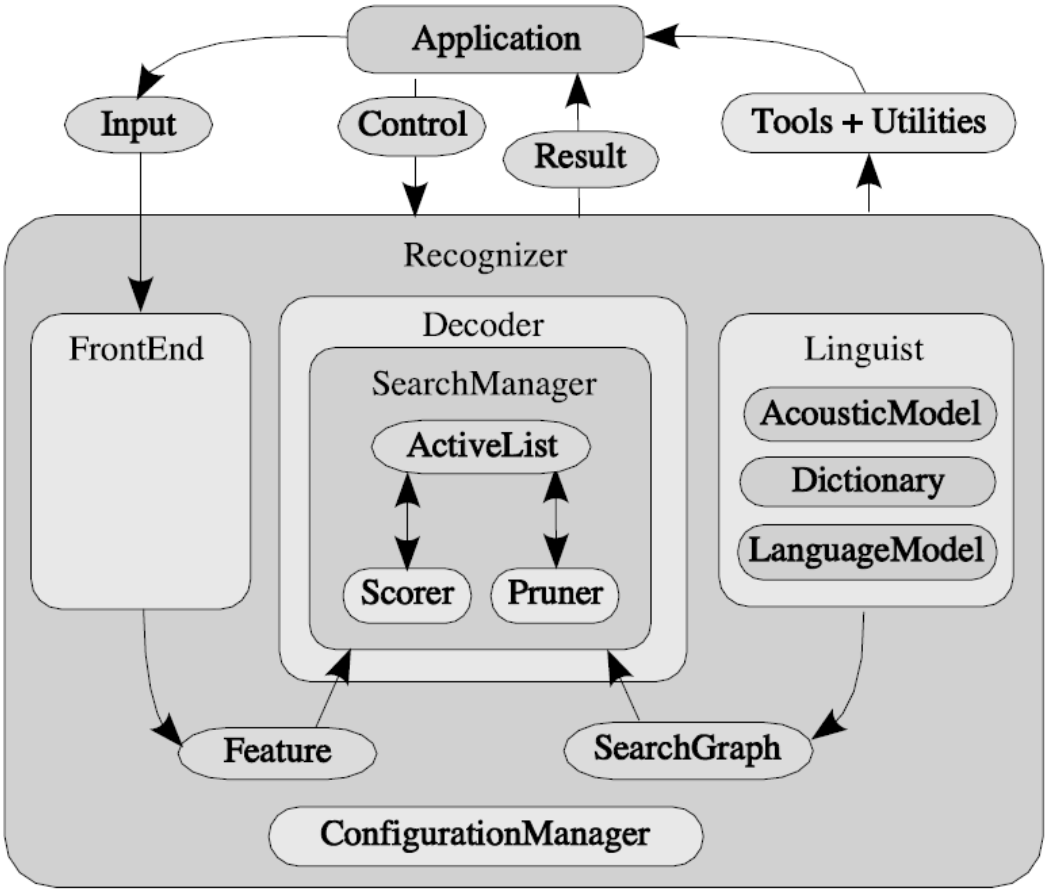


Figure 3-2: The Sphinx4 Framework (Sphinx 4 White Paper)

Each module may be implemented by different classes, each with its own set of parameters, allowing many different ways to use and configure Sphinx. For this project, Sphinx was configured for large-vocabulary continuous speech recognition, using the following modules:

- FrontEnd
  - audioFileDataSource, dataBlocker, speechClassifier, speechMarker, nonSpeechDataFilter, premphasizer, windower, fft, melFilterBank, dct, batchCMN and featureExtraction

- Decoder
  - WordPruningBreadthFirstSearchManager
  - ThreadedAcousticScorer
  - SimplePruner

- LexTreeLinguist
    - TiedStateAcousticModel
    - FastDictionary
    - LargeNGramModel

The FrontEnd manages the audio input source and pipeline, and is configured to read a WAV file with a single audio channel encoded with pulse-code modulation (PCM) at 16 KHz.

The LexTreeLinguist manages the acoustic and language models. The LargeNGramModel class is used, configured for a trigram model. The dictionary and language model approach used in this architecture constrain the recognition process to a single-word vocabulary model (for example, "process" and "processing" are distinct words). This means that the dictionary and language model must contain all word variants to be recognized.

The Decoder generates recognition hypotheses and results. The Decoder configuration can have a significant impact on performance and accuracy, for example by increasing or decreasing the search space and number of hypotheses evaluated. In general, accuracy may be improved at the expense of performance (the recognizer requires more memory and is slower). As this project investigates the relative performance of competing language models, reasonable defaults were used for the Decoder but further configuration-related optimizations were not explored.

The full configuration of Sphinx4 used is included as Appendix 4.

3.6 Reference acoustic and language models

The reference acoustic and language models used are the HUB4 models provided with Sphinx.

The HUB4 Acoustic Models are US English models generated from 140 hours of audio from the 1996/7 HUB4 corpus. The specific model used is an 8-gau, 6000 senone tri-state HMM (packaged in hub4opensrc.cd_continuous_8gau.zip) [76]. The model uses the cmudict_0.6d phoneset without stress markers and a silence phone, totalling 40 phones.

The HUB4 Language Model (packaged in HUB4_trigram_lm.zip) is a trigram LM generated from "a variety of permitted sources, including broadcast news" [76] with a vocabulary of 64000 words.

The models produce reasonable word error rates within the reported range for Sphinx4 when used for continuous speech recognition of US English speakers.

3.7 Selection of lectures

As the project investigates the performance of different language models, sample lectures were selected with the goals of minimizing the influence of extraneous variables on the recognition process, while ensuring a reasonable spread of topics and speakers.

Requirements for sample lectures were thus:

- Good-quality audio (recorded with a close-talking microphone, minimal reverberation or background noise)
- Speakers with a North American English accent (likely to be a reasonable match with the reference acoustic model)
- Lectures should be from a higher education institution on a range of topics (matching the application domain)
- Lectures should be in the form of a continuous monologue, thus no or little audience interaction or third-party media such as film clips (to reduce the impact of different speakers or variable quality audio)

The Open Yale Courses (OYC) site was identified as a suitable collection containing many lectures matching the above requirements, and helpfully includes transcripts for all lectures. Audio recordings and transcripts are licensed with a Creative Commons Attribution Non-Commercial ShareAlike license which facilitates their use in research applications and the downstream publication of derivative works such as modified transcripts.

A subset of OYC lectures was selected to ensure a diversity of knowledge domains, a range of speakers and recordings of a consistent length (approximately 50 minutes). A subjective listening test was used to further select recordings with the best audio quality.

Assessing the performance of language models across different domains and topics is considered important because some disciplines use many more specialist words than others, which is likely to affect recognition and thus search performance.

Table 3-1 shows the final set of 13 selected lectures. (References for each are listed in Appendix 3).

#	Course	Lecture title	Lecturer
1	ASTR 160: Frontiers and Controversies in Astrophysics	Dark Energy and the Accelerating Universe and the Big Rip	Professor Charles Bailyn
2	BENG 100: Frontiers of Biomedical Engineering	Cell Culture Engineering	Professor Mark Saltzman
3	BENG 100: Frontiers of Biomedical Engineering	Biomolecular Engineering: Engineering of Immunity	Professor Mark Saltzman
4	EEB 122: Principles of Evolution, Ecology and Behavior	Mating Systems and Parental Care	Professor Stephen Stearns
5	ENGL 220: Milton	Lycidas	Professor John Rogers
6	ENGL 291: The American Novel Since 1945	Thomas Pynchon, The Crying of Lot 49	Professor Amy Hungerford
7	ENGL 300: Introduction to Theory of Literature	The Postmodern Psyches	Professor Paul Fry
8	HIST 116: The American Revolution	The Logic of Resistance	Professor Joanne Freeman
9	HIST 202: European Civilization, 1648-1945	Maximilien Robespierre and the French Revolution	Professor John Merriman
10	PHIL 176: Death	Personal identity, Part IV; What matters?	Professor Shelly Kagan
11	PLSC 114: Introduction to Political Philosophy	Socratic Citizenship: Plato, Apology	Professor Steven Smith
12	PSYC 110: Introduction to Psychology	What Is It Like to Be a Baby: The Development of Thought	Professor Paul Bloom
13	RLST 152: Introduction to New Testament History and Literature	The "Afterlife" of the New Testament and Postmodern Interpretation	Professor Dale Martin

Table 3-1: Selected Open Yale Courses lectures

3.8 Recognition process with reference language model

Figure 3-3 illustrates the process followed to execute the speech recognition process for a recorded lecture and the reference HUB4 language model.

The audio is downloaded from the OYC collection and converted from the published mp3 format to the 16 KHz mono WAV format required by Sphinx. The lecture transcript is downloaded and conditioned into a continuous set of unpunctuated words as the reference transcript.

Sphinx is configured with the HUB4 acoustic model, HUB4 LM and accompanying dictionary as described in 3.5 and Appendix 4, and then run with the input audio file, producing a hypothesis transcript. The reference and hypothesis transcripts are then compared and evaluated to generate the metrics used for analysis such as WER.

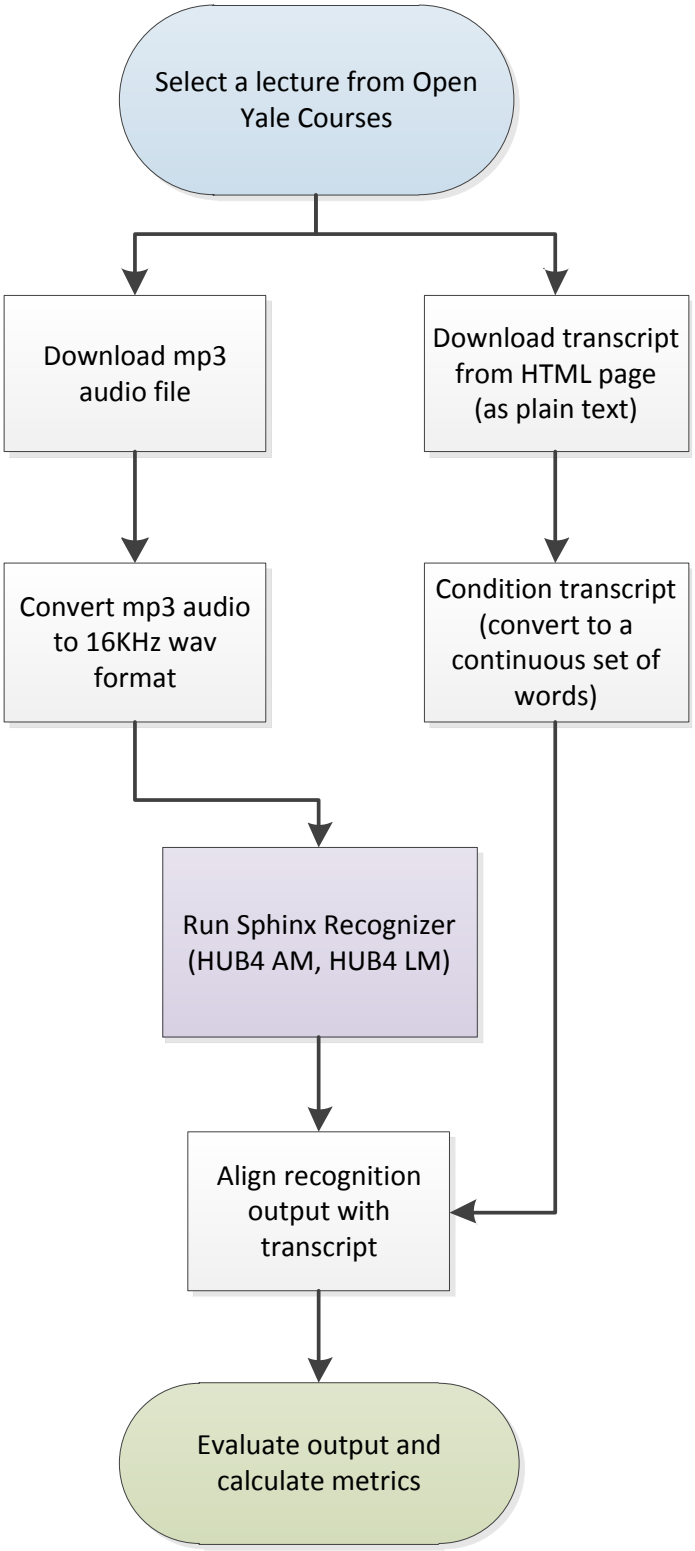


Figure 3-3: Recognition process with a reference language model

3.9 Calculating metrics

As described in 3.3, three primary and four secondary metrics are used to evaluate the results of the techniques applied.

Primary metrics are:

- Word Error Rate (WER)
- Word Correct Rate (WCR)
- Ranked Word Correct Rate (RWCR)

Secondary metrics are:

- Out of vocabulary (OOV) words, representing vocabulary coverage
- Unrecognized words, representing vocabulary recognition
- Extraneous words, representing vocabulary recognition
- Language model perplexity, representing the complexity and alignment of the language model in relation to the target text.

The metrics are calculated from the reference transcript, hypothesis transcript (as produced by the recognizer), language model, and in the case of RWCR also a frequency-ranked English dictionary, as shown in Table 3-2. It is assumed that the pronunciation dictionary is equivalent to or a superset of the language model's vocabulary.

Metric / Resource	Reference transcript	Hypothesis transcript	Language model	Frequency-ranked dictionary
WER	✓	✓		
WCR	✓	✓		
RWCR	✓	✓		✓
OOV words	✓		✓	
Extraneous words	✓	✓		
Unrecognized words	✓	✓		
Perplexity	✓		✓	

Table 3-2: Recognition metrics and artefacts

Word Error Rate is calculated as:

$$WER = \frac{\mathrm{S} + \mathrm{D} + \mathrm{I}}{\mathrm{N}}$$

where S = substitutions, D = deletions, I = insertions, N = word count of reference transcript. Calculating S, D and I requires the reference and hypothesis transcripts to be aligned, as illustrated in Table 3-3:

Reference	Hypothesis
The best way, I think, to introduce the central issues of this wonderful poem, *Lycidas,* is to return to Milton's *Comus*. So yet once more -- and I promise this will be one of the last times that we look back at Milton's mask -- but yet once more, let's look at *Comus.* Now you will remember that the mask *Comus* was everywhere concerned with questions of the power of -- well, the strangely intertwined questions of the power of chastity on the one hand and the power of poetry on the other.	the best way to buy thank to introduce the central issues of of it's a wonderful column was so this is is to return set to milkens common so we can once more i promise this will be one of the last times that we look back at hilton's masked but what's yet once more let's look at our comments making remember now the mass comments was everywhere concerned with questions of the power of well because strangely intertwined questions of the power of chassis on the one hand the power of poetry on the other
Aligned Reference	**Aligned Hypothesis**
`the best way ** I THINK to introduce the central issues ** of **** THIS wonderful ****** *** ** POEM LYCIDAS is to return *** to MILTON'S COMUS so ** YET once more AND i promise this will be one of the last times that we look back at MILTON'S MASK but ****** yet once more let's look at COMUS NOW YOU WILL remember THAT the MASK COMUS was everywhere concerned with questions of the power of well THE strangely intertwined questions of the power of CHASTITY on the one hand AND the power of poetry on the other`	`the best way TO BUY THANK to introduce the central issues OF of IT'S A wonderful COLUMN WAS SO THIS IS is to return SET to MILKENS COMMON so WE CAN once more *** i promise this will be one of the last times that we look back at HILTON'S MASKED but WHAT'S yet once more let's look at OUR COMMENTS MAKING **** remember NOW the MASS COMMENTS was everywhere concerned with questions of the power of well BECAUSE strangely intertwined questions of the power of CHASSIS on the one hand *** the power of poetry on the other`
'***' = Hypothesis inserted a word incorrectly Upper case = word substituted.	'***' = Hypothesis deleted a word incorrectly Upper case = insertion or substitution.

Table 3-3: Reference and hypothesis transcripts with alignment

In the above example, the reference transcript contains 90 words, and the recognition hypothesis has 18 substitutions, 3 deletions and 9 insertions. The Word Error Rate is thus (18 + 3 +9) / 90 = 33.3%.

Word Correct Rate is calculated as the number of correct words as a proportion of total word count, so in the above example WCR is 69 / 90 = 76.6%.

Here WCR is higher than the inverse of WER, as WER reflects errors by the recognizer in identifying whether a phoneme sequence constitutes one or two words. Where the hypothesis has significantly fewer or more words than the reference, WER will also diverge further from inverse of WCR.

The transcript alignment and resulting WER and WCR metrics are calculated by the NISTAlign class from CMU Sphinx4.

The RWCR, OOV, extraneous and unrecognized words metrics are calculated with custom-written scripts (source code links are provided in Appendix 2). Table 3-4 shows examples of OOV, extraneous and unrecognized words using the above example.

Transcript vocabulary	OOV words	Extraneous words	Unrecognized words
Comus, I, Lycidas, Milton's, and, at, back, be, best, but, central, chastity, concerned, everywhere, hand, intertwined, introduce, is, issues, last, let's, look, mask, more, now, of, on, once, one, other, poem, poetry, power, promise, questions, remember, return, so, strangely, that, the, think, this, times, to, was, way, we, well, will, with, wonderful, yet, you	comus milton's lycidas	comments what's thank set our milkens mass masked making it's hilton's common column chassis can buy because a	mask and you think poem chastity
54	**3**	**18**	**6**

Table 3-4: Example of transcript vocabulary, OOV, extraneous and unrecognized words

OOV words are those occurring in the transcript but not in the language model, reflecting limitations in the language model vocabulary.

Extraneous words are those which occur in the hypothesis transcript, but are not in the reference transcript. These may reflect recognition difficulties, or a language model which has too large a vocabulary or is too diverse.

Unrecognized words are those which are in the language model and the reference transcript, but not in the hypothesis. These may reflect audio-related recognition difficulties (for example from background noise) or language-related recognition difficulties arising from poor alignment between the language model and the genre, style of speech or topic of the recorded speech.

The perplexity of the language model in relation to the reference transcript is calculated by the evaluate-ngram tool in the mitlm language modelling toolkit.

Finally, the Ranked Word Correct Rate is calculated by taking into account the dictionary rank of each word and including recognition scores only for those words below a given frequency cut-off, as illustrated in Table 3-5 for a cut-off rank of 10,000:

Word	Dictionary frequency rank	Recognized	Unrecognized
STRANGELY	17238	1	0
INTERTWINED	25037	1	0
CHASTITY	29904	0	1
MILTON'S	41755	0	2
COMUS	91192	0	3
LYCIDAS	157200	0	1
Total		**2**	**7**

Table 3-5: Calculation of Ranked Word Correct Rate

Here only the recognition rates of words which occur below the dictionary frequency rank cut-off value are considered. The RWCR for this example is thus 2 / (2+7) = 22.2%.

This reflects the application-specific assumption that search terms are more likely to be less common words (for example “Comus” rather than “everywhere”) and therefore these words are more valuable for recognition.

4 Topic and language modelling with Wikipedia

4.1 Introduction

This chapter introduces Wikipedia as a linguistic resource, and describes the process used for converting a set of Wikipedia articles into a plain text corpus suitable for generating or adapting a language model. Topic modelling in Wikipedia is then introduced.

A technique is described for identifying and harvesting a set of related Wikipedia articles using article similarity metrics generated through latent semantic indexing, enabling the creation of topic-specific custom language models.

4.2 Wikipedia as a linguistic resource

The English Wikipedia (hereafter Wikipedia) is used to create three types of resources for this project:

1. a dictionary of English words with word frequency counts
2. a generic language model, approximating general English usage
3. topic-specific language models, approximating English usage in a topic area

Advantages of using Wikipedia for this purpose include:

- It is a large corpus, containing more than 4 million articles and over 1000 million words (although other language versions of Wikipedia are smaller) [77], [78]. It is thus of a similar order of magnitude to resources such as the English Gigaword Corpus [79].
- It has been shown to be a usable language resource for other natural language processing tasks [80].
- Wikipedia articles include semantic metadata through inter-article links and other tags such as categories. This semantic structure can be used to select subsets of Wikipedia articles.
- It has broad topic coverage.
- It is updated continuously, and thus dynamic and contemporary.
- Wikipedia text is available at no cost, and published with a permissive license allowing derivative works to be freely redistributed [81].

The principle disadvantage is that it is a loosely curated resource, and thus contains a greater number of typographical, spelling, formatting and classification variations and errors than other published texts which have been edited in a more traditional and centralized manner. For applications such as this one which make use of Wikipedia as source data for statistical models, these types of errors are less significant, provided they are of relatively low frequency.

4.3 Creating a plain text corpus from Wikipedia

Users interact with Wikipedia as a set of article web pages, for example as shown in Figure 4-1. Each page contains global navigation links, links to article metadata such as the history and discussion pages, links to other articles within the article body text, and reference information such as footnotes.

To create a plain text corpus, only the actual body text is of interest. Wikipedia articles are stored in wiki markup format rather than HTML, illustrated in Figure 4-2. Wiki markup is preferable as a source format for further processing because the wiki markup typically has more semantic value than the equivalent HTML representation and can thus be processed more reliably.

Figure 4-1: Lycidas Wikipedia article, as shown in a web browser

```
{{For|the genus of jumping spiders|Lycidas (genus)}}

'''"Lycidas"''' is a poem by [[John Milton]], written in 1637 as a [[pastoral]] [[elegy]]. It first appeared in a 1638 collection of elegies, entitled ''Justa Edouardo King Naufrago'', dedicated to the memory of [[Edward King (British poet)|Edward King]], a collegemate of Milton's at [[Cambridge]] who drowned when his ship sank in the [[Irish Sea]] off the coast of Wales in August 1637. The poem is 193 lines in length, and is irregularly rhymed. While many of the other poems in the compilation are in Greek and Latin, "Lycidas" is one of the poems written in English.<ref>Womack, Mark. "On the Value of Lycidas." Studies in English Literature, 1500-1900 1997: [119-136]. JSTOR. 3 Nov 2008 <http://www.jstor.org/stable/450776></ref> Milton republished the poem in 1645.
```

Figure 4-2: Lycidas Wikipedia article wiki markup text

However, for continuous speech recognition language modelling purposes where the model should be trained on sentences approximating how people speak, punctuation and references are unwanted, and so further text conditioning (the process of converting text to a consistent, canonical form) is applied to transform wiki markup into a list of unpunctuated, upper-case sentences, illustrated in Figure 4-3.

LYCIDAS IS A POEM BY JOHN MILTON WRITTEN IN 1637 AS A PASTORAL ELEGY

IT FIRST APPEARED IN A 1638 COLLECTION OF ELEGIES ENTITLED JUSTA EDOUARDO KING NAUFRAGO DEDICATED TO THE MEMORY OF EDWARD KING A COLLEGEMATE OF MILTON'S AT CAMBRIDGE WHO DROWNED WHEN HIS SHIP SANK IN THE IRISH SEA OFF THE COAST OF WALES IN AUGUST 1637

THE POEM IS 193 LINES IN LENGTH AND IS IRREGULARLY RHYMED

WHILE MANY OF THE OTHER POEMS IN THE COMPILATION ARE IN GREEK AND LATIN LYCIDAS IS ONE OF THE POEMS WRITTEN IN ENGLISH

MILTON REPUBLISHED THE POEM IN 1645

Figure 4-3: Conditioned sentences from the Lycidas Wikipedia article

This list of sentences provides the source material from which a language model and dictionary with word frequency counts can be generated.

Figure 4-4 illustrates the complete process of creating a plain text corpus from Wikipedia [82].

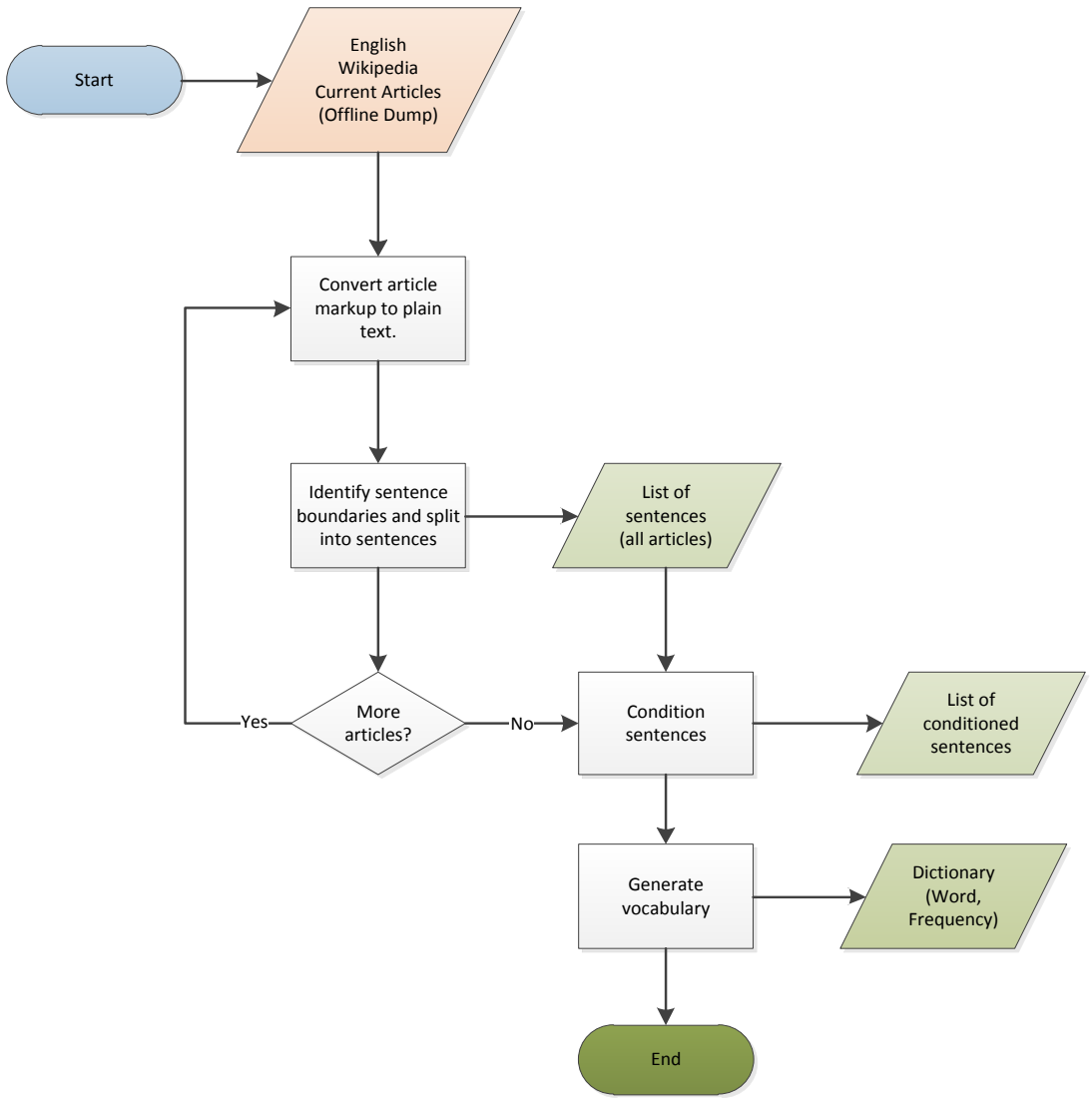


Figure 4-4: Generating a plain text corpus from Wikipedia

Article markup text is available from Wikipedia in two forms:

- for an individual article, through the Wikipedia API
- for a set of articles, as a compressed XML dump file containing a snapshot of all articles and their metadata.

The most efficient method to access the body text for a large set of articles is to download and process the dump file. For this process, a dump of the current revision of all articles labelled "enwiki-latest-pages-articles" is used.

The gwtwiki toolkit is used to parse the Wikipedia XML dump and extract the article titles and wiki markup text. (Further details of the software toolkits and datasets used are contained in Appendix 1.)

The following text conditioning is applied to each article to produce a list of sentences:

- Headings and references are removed
- gwtwiki's PlainTextConverter is used to render the remaining wiki markup to plain text (removing, for example, markup used for inter-article links)
- The OpenNLP toolkit is used for statistical sentence boundary detection. This is more reliable than using a regular-expression parser, as English punctuation can be ambiguous (for example a full-stop may be used for abbreviations within a sentence).
- A further set of rules is applied to restrict the output set as far as possible to well-formed sentences:
    - sentences must contain 5 or more words
    - sentences must start with a capital letter (A-Z) and end with a full-stop, question mark or exclamation mark
    - sentences containing "/" or "&" are rejected (which excludes URLs)
- Sentences are capitalized and punctuation is removed.

A further word filter is applied when the dictionary and word counts are generated:

- Words must satisfy English orthography (consisting only of letters a-z, hyphen and apostrophe) and English apostrophization rules.
- Words must be three letters or longer.

The latter condition of course excludes many valid English words (a, an, to), which are restored to the resulting dictionary from a list of 2-letter words contained in the CMU Dictionary.

The above rules filter out many non-words and non-sentences in Wikipedia articles, introduced both through meaningful content such as text in tables, bulleted lists and abbreviations, and through misspellings or syntactic errors.

The filtering process compensates to a degree both for noise in the data, and at a higher level for the fact that Wikipedia as a written, visual, hypertext genre is being used to model language use in continuous speech, a linear, oral genre.

Two language resources are created from the above process: a large set of conditioned, plain text upper-case sentences from all articles (an extract of which is shown in Figure 4-4), and a frequency-ranked word dictionary.

4.4 Goals for the adapted language model

When creating a custom language model adapted to a specific topic, the goal is not necessarily to create a larger model, but to create a well-adapted model, that is a model which is closely aligned to the recognition target text in genre, vocabulary, linguistic style and other dimensions.

The size of an n-gram language model is initially determined by the number of different n-grams (combinations of n words) encountered in the training text. Models may be limited in size by:

- constraining the vocabulary, in which case words in the training text which are not in the given dictionary will be modelled as "unknown", and
- applying a frequency cut-off to the n-grams, in which case n-grams which occur fewer than a certain number of times in the training text will not be included in the model.

In general, a larger language model increases the search space for the recognizer, and for the Sphinx4 recognition engine, larger models lead to an increase in both runtime (a consequence of the larger search space) and memory requirements (a consequence of needing to load the entire model into memory).

A further consequence of increasing the search space with a larger model is that accuracy can be reduced as the model leads to the recognizer introducing extraneous words and phrases.

To enable the most accurate comparison between recognition performance with the adapted and reference language models, the adapted models are created with the same vocabulary size as the HUB4 reference model, 64000 words.

However, owing to limitations in the language modelling toolkit used, a frequency cut-off was not applied to the adapted language model. This leads to the adapted model having a larger number of bi-grams and tri-grams than the reference model, and overall being about twice the size as presented in section 5.4 (Table 5-6).

4.5 Constructing a topic-adapted language model from Wikipedia

Figure 4-5 illustrates the steps taken to construct a topic-adapted language model from Wikipedia.

In Step 1, a set of articles is identified from Wikipedia which relate to the topic keywords. The article selection process is described further in 4.8 to 4.11.

In Step 2, the output text from each of the articles is conditioned into plain text sentences using the techniques described in 4.3.

Steps 3 and 4 create a target vocabulary for the adapted language model. This is done by merging two frequency-ranked vocabularies: one, the more specialized, derived from the output text from the selected set of Wikipedia articles, and the other, more general, derived from a plain text corpus of all Wikipedia articles. The merged list starts with all words which occur 5 or more times in the specialized word list, and is supplemented with words from the general list in descending order of frequency until the list reaches the target size of 64,000.

For the word cutoff threshold and a number of other parameters (for example the Wikipedia crawler constraints listed in Table 4-2, and the similarity threshold applied in 4.11), reasonable default values have been chosen informed by experimental results. Further experimentation would be required to establish the impact of varying these choices and whether the selected values are optimal.

Step 5 creates a phonetic dictionary for the target vocabulary, described further in 4.6.

The adapted language model is then created in Steps 6 and 7, using the mitlm language modelling toolkit.

As the amount of training text available from the set of topic-related Wikipedia articles is relatively small, a more general language model is first created from a larger Wikipedia corpus, restricted to the target vocabulary (Step 6a). The input corpus used for this model is 5% of all Wikipedia text, selected using 1 from every 20 sentences, yielding a total of around 75 million words.

A topic-specific language model is then created from the conditioned text output from the topic-related Wikipedia articles, again restricted to the target vocabulary (Step 6a).

The two language models are then merged using linear interpolation to create the third, adapted language model (Step 7).

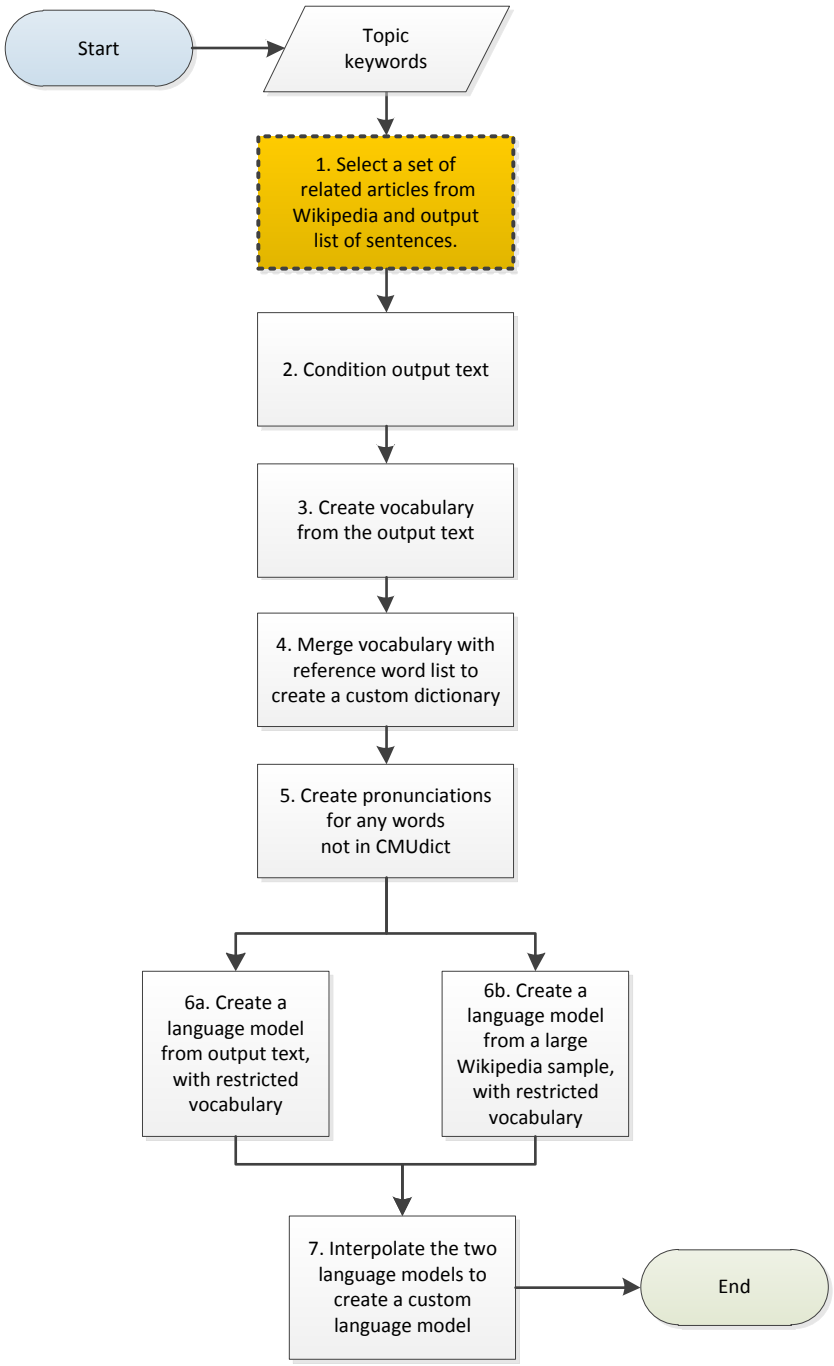


Figure 4-5: Creation of a custom language model from Wikipedia

4.6 Recognition process with a custom language model

Figure 4-6 illustrates the process followed to execute the speech recognition process for a recorded lecture with a custom, adapted language model.

The method of language model adaptation here is unsupervised adaptation based only on minimal information about the lecture, in the form of up to 5 keywords derived from the lecture topic. It is assumed that a suitable set of keywords could always be selected, possibly in an automated way, from the subject area of the lecture (for example the from the name of the department and title of the course) and the title of the lecture.

Example keywords following this approach for three lectures are shown in Table 4-1.

Course	Lecture	Keywords
ASTR 160: Frontiers and Controversies in Astrophysics	Dark Energy and the Accelerating Universe and the Big Rip	astrophysics dark energy accelerating universe
ENGL 220: Milton	Lycidas	english literature milton lycidas
PLSC 114: Introduction to Political Philosophy	Socratic Citizenship: Plato, Apology	political philosophy socratic citizenship plato

Table 4-1: Examples of keywords for selected lectures

The custom language model is created automatically as described in 4.8 to 4.11, seeded by the given keywords. The recognition process in all other respects is the same as that for the reference language model (3.8).

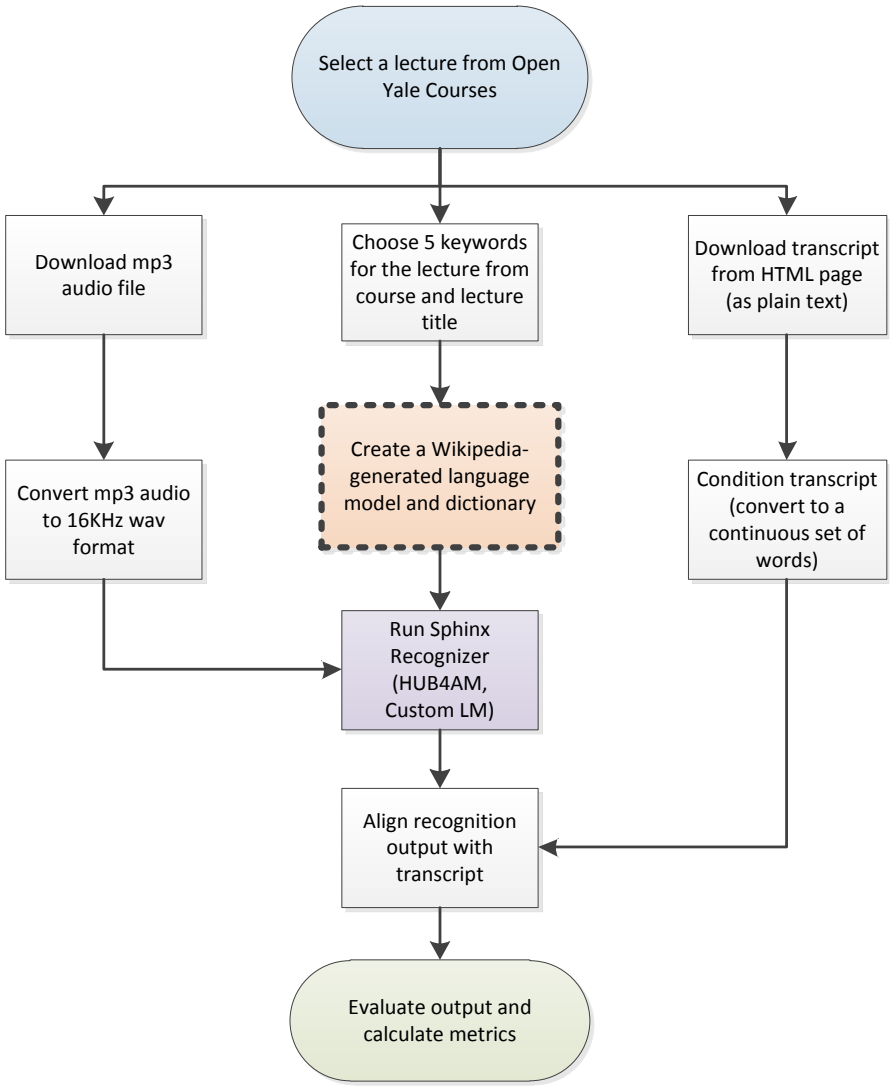


Figure 4-6: Recognition process with a custom language model

4.7 Constructing the phonetic dictionary

The base phonetic dictionary used is the CMU Pronouncing Dictionary (CMUDict) 0.7a, which contains phonetic representations for slightly over 123,000 words. However, it is to be expected that new words will be encountered which are not in the CMU Dictionary, for example because of their relative scarcity, or because they are neologisms or variants of known words such as new hyphenations.

As less common words are expected to be significant to the topic, it is important to recognize them where possible, and thus a method is required to generate pronunciations for unknown words.

For this project, the phonetisaurus grapheme-to-phoneme (g2p) converter is used. Phonetisaurus uses a weighted finite state transducer (WFST) approach to generate pronunciation hypotheses for a word, a technique claimed to produce results comparable in accuracy to other state-of-the-art systems [83].

The model used by phonetisaurus for this application is trained from CMUDict 0.7a and thus phonetisaurus is in effect extrapolating from the implicit pronunciation rules represented in CMUDict. Only the best hypothesis generated by phonetisaurus is used.

Stress markers are used in the training process to create the FST model, but ignored in the output. This approach is supported by experimental results suggesting that Sphinx recognition accuracy is adversely affected by using stress markers, but that g2p models trained from CMUdict with stress markers produce better accuracy even when stress markers in the output are ignored [76], [84].

4.8 Identifying a topic-related subset of Wikipedia articles

To collect a set of articles from Wikipedia related to the topic of a lecture, a web crawler strategy is used. A web crawler starts at a seed page and iteratively follows all the links encountered on the page.

In this case, the crawler is seeded with the top five search results produced by executing a Wikipedia search using the keywords selected from the topic title. The crawler then follows a breadth-first search, by adding links it encounters on each page to the end of an article queue, and iterating through the article queue. Only links to other Wikipedia articles are followed; the crawler thus does not follow links to external sites.

Two different crawler strategies are investigated:

1. The Naïve Crawler (4.9) follows all article links it encounters on the page.

2. The Similarity Crawler (4.11) is more discriminating in which article links it follows, and only follows those for which the target article is sufficiently similar to the current article, using an article similarity metric defined in 4.10.

The crawlers are bound by a number of constraints to ensure that they terminate with a reasonable set of output text, and will not visit the same article more than once.

As Wikipedia articles typically contain links to both related and unrelated topics, it is expected that the set of articles indexed by the Naïve Crawler would cover a broader range of topics than those indexed by the Similarity Crawler. The effectiveness of the similarity measure can therefore be evaluated by comparing the recognition performance of a language model adapted using the naïve crawler's output text with one adapted using the Similarity Crawler's output text.

4.9 The Wikipedia Naïve Crawler

The operation of the naïve crawler is illustrated in Figure 4-7, with the set of parameters used to limit its operation listed in Table 4-2.

In Step 1, a Wikipedia search is executed using the Wikipedia Search API (https://en.wikipedia.org/w/api.php), and the first five articles meeting the minimum seed article word count are added to the search queue (Step 2).

The crawler then processes the article at the top of the search queue (Step 3) until an exit condition is met. Exit conditions are the maximum search depth has been reached (at most 5 links from the seed article to the indexed article), the maximum number of articles to index has been reached (2500 articles), or the maximum number of output sentences has been reached (200 000 sentences).

The full text of the article is retrieved in wiki markup format.

In Step 4, the markup text is conditioned to produce a list of plain text sentences, which is appended to the sentence output file following the same conditioning process described in 4.3.

In Step 5, the markup text is parsed to extract the set of links to other articles not already visited, and in Step 6, the titles of articles not already visited are added to the search queue.

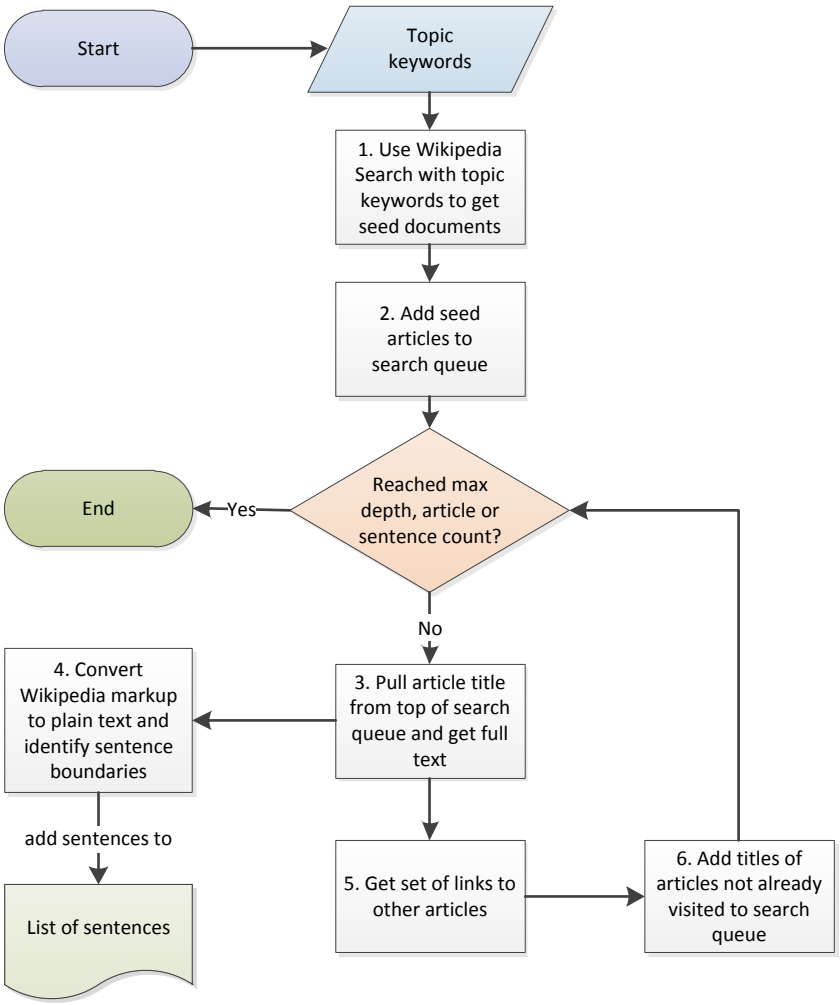


Figure 4-7: Wikipedia crawler with naïve strategy

Parameter	Value
Maximum seed pages from search results	5
Minimum seed article word count	250
Maximum article depth	5
Maximum number of articles	2500
Maximum number of output sentences	200 000

Table 4-2: Wikipedia crawler constraints

Figure 4-8 illustrates the naïve crawler in action for a sample lecture.

Four keywords are chosen: “english”, “literature”, “milton” and “lycidas”. At depth 0, the seed article “Lycidas” includes links to both “John Milton” (more relevant) and “Cambridge” (less relevant). Topic coverage then diverges as search depth increases.

The crawler exits when the maximum output sentence count is exceeded at 200 156 sentences, having processed 2335 articles at a maximum depth of 2.

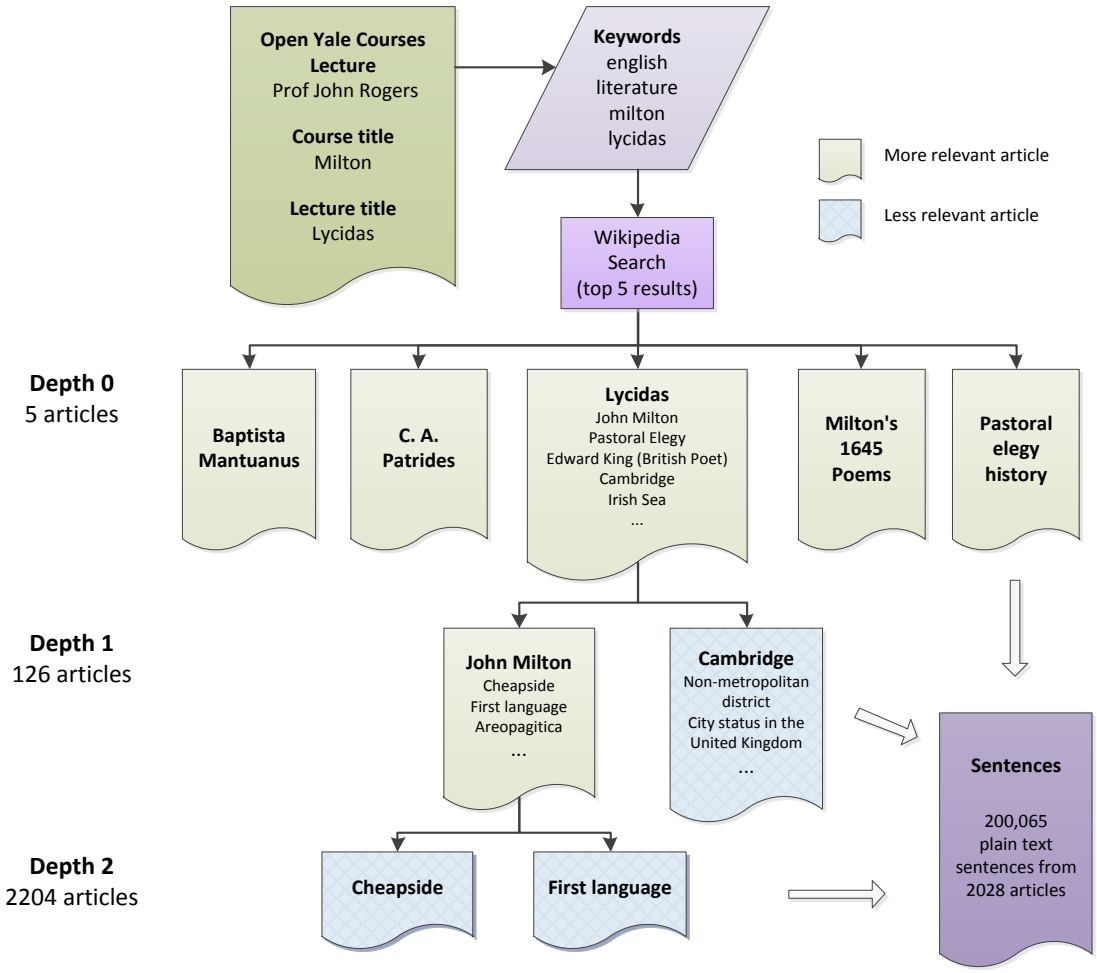


Figure 4-8: Wikipedia Crawler for lecture on Lycidas (naïve crawler)

4.10 Topic modelling with article similarity

When adapting a language model to a given topic, two goals are:

- to improve the vocabulary coverage of the model for the given topic, i.e. to include as many words as possible which are likely to be used in the context of the topic, and
- to model the style of language and typical word combinations used in the context of the topic.

It is therefore advantageous to collect as much text as possible from contexts (here, Wikipedia articles) which are related to the target topic. And as this process should be unsupervised, it must be possible to establish "relatedness" in an automated way without human subjective judgement or interpretation.

This section describes an article similarity metric which gives the degree of similarity (measured from 0 to 1) between two Wikipedia articles. This metric is then used to improve the discrimination of a Wikipedia article crawler, such that only similar articles are included in the links which are followed. This approach, described further below, aims to gather a large set of articles using search seeding and transitive similarity.

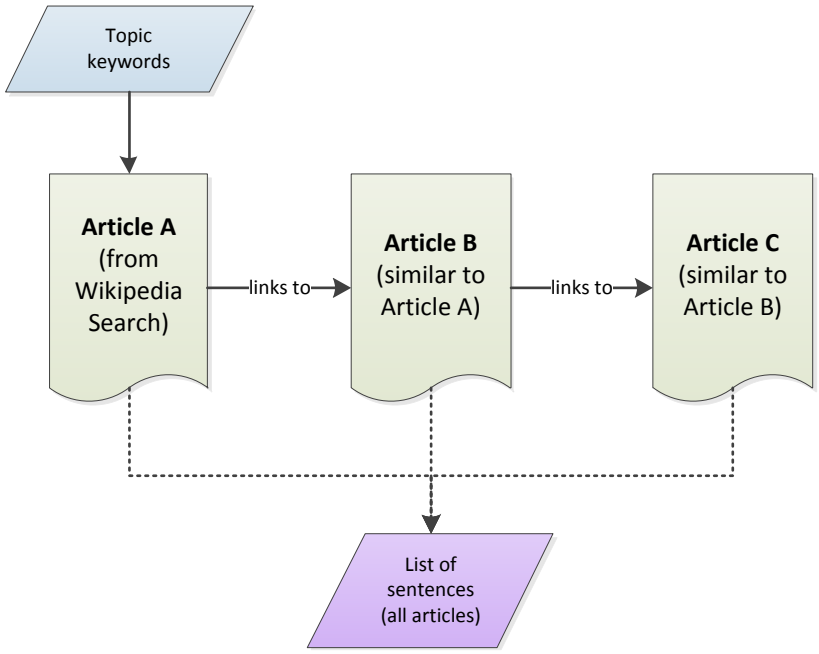


Figure 4-9: Seeded search with transitive similarity

As shown in Figure 4-9, a search is seeded using keywords, with subsequent articles being included in the search net through similarity to the parent article: Article B is similar to Article A, and Article C is similar to Article B. The text from all such articles is then used to train a language model for the target topic.

Latent semantic indexing (LSI) is used to derive an article similarity metric. LSI, also known as Latent Semantic Analysis (LSA), is a technique widely used in information retrieval applications to identify related documents in large corpora [85], [86]. LSI uses singular value decomposition to train a model from the corpus which relates individual words to a set of topics. The set of topics is of a fixed-size and arbitrary (in that the topics are mathematical abstractions which emerge from latent semantic clustering in the data). Each topic is defined through a set of words and their respective contribution weights to the topic. A related method, Latent Dirichlet Allocation (LDA), works in a similar way but is not explored here.

Using the model, a document may then be expressed as a set of topic values (representing the relative strength of each topic in the document), or equivalently as a set of n values representing a position in n-dimensional space (where n is the number of

topics in the model). The similarity between two articles is then understood to be the distance between the two article vectors in an n-dimensional space.

To apply LSI to Wikipedia and generate article similarity scores, the open source gensim vector space modelling toolkit is used [87]. gensim is designed to handle large corpora such as Wikipedia which exceed available memory, and in addition is well-documented and actively maintained.

Figure 4-10 illustrates the initial process to train the LSI model from Wikipedia, a modified version of the recipe described in the gensim documentation [88].

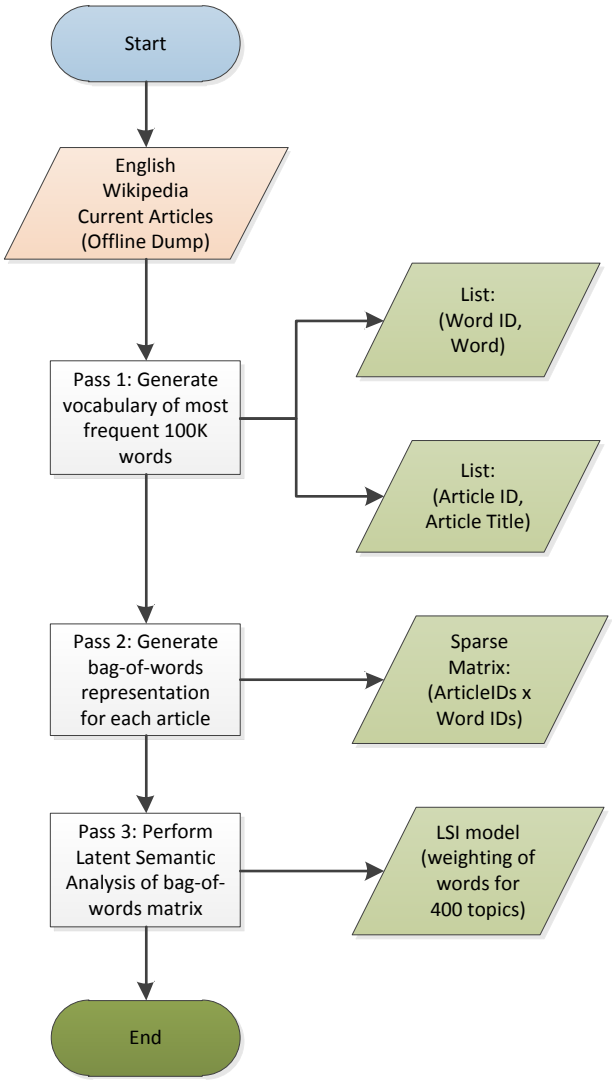


Figure 4-10: Generating a bag of words index and LSI model from Wikipedia with gensim

This requires three passes through an offline dump of all Wikipedia articles (3,345,476 articles in total from the Wikipedia snapshot used). This is a time-consuming process, but only needs to be executed at the start (and possibly at intervals thereafter to take account of gradual evolution of the corpus).

In Pass 1, a vocabulary is created of the most frequent 100,000 words. Only these words will be regarded as significant for LSI modelling, with any remaining words or tokens being ignored. Outputs from this pass are the list of words (each with a numeric identifier), and list of article titles (also each assigned a numeric identifier).

In Pass 2, a bag-of-words representation of each article is generated. This represents the article as the set of distinct words from the chosen vocabulary which occur in it (irrespective of frequency, word position or sequencing). This is the simplest article representation which can be used with this technique; other approaches such as using a TF-IDF measure can be used but are not explored here. The output of this pass is a sparse matrix of words by articles.

In Pass 3, the sparse matrix is used to create the LSI model for a fixed number of topics, 400.

The model parameters of 100,000 terms and 400 topics follow the gensim defaults, informed by empirical results on dimensionality for semantic indexing applications suggesting an optimal range of 300 to 500 for topic size [89].

Figure 4-11 illustrates the process followed to generate scores for similarity between a parent article and one or more linked articles using gensim with the LSI model.

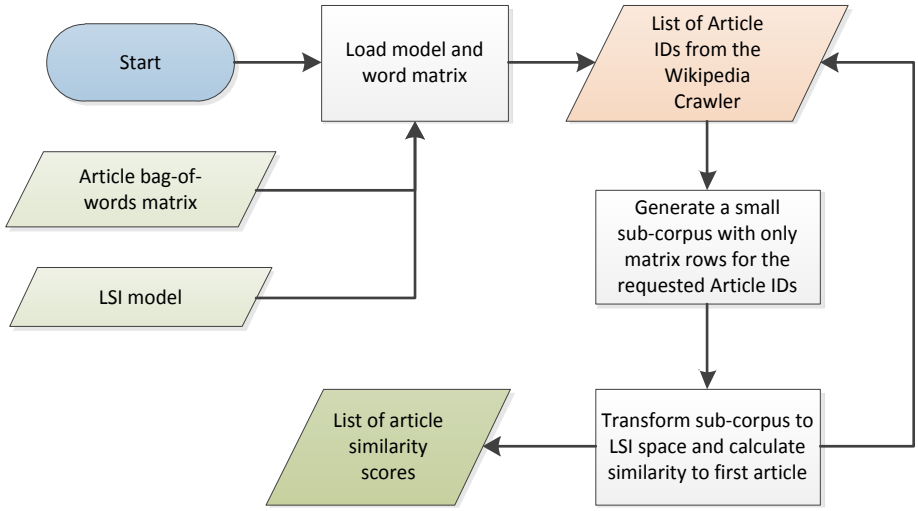


Figure 4-11: Generating article similarity scores with gensim and an LSI model

While the initial creation of the LSI model from Wikipedia is time-consuming (upwards of 24 hours), calculating similarity between a document and the set of documents to which it links is relatively efficient, at approximately 72ms per comparison. This makes it a computationally tractable approach for generating custom language models on demand, requiring neither a significant memory footprint nor long runtime.

By comparison, pre-computing pair-wise article similarity for the approximately 3.3 million articles in the Wikipedia snapshot would require a set of 5.6×10^{12} tuples, which would take just under 13,000 processor-years to calculate.

For efficiency, the similarity calculation engine is designed to execute as a long-lived process with which the crawler communicates, so that the LSI model and word matrix are only loaded once, rather than per article or per comparison.

4.11 The Wikipedia Similarity Crawler

Figure 4-12 illustrates the operation of the Wikipedia Similarity Crawler.

This resembles the Naïve Crawler described in 4.9 with the addition of the Similarity Scorer in Step 6. Here the crawler passes a list of articles to the scorer, which calculates and returns a set of similarity scores for the target articles, ranging from 0 (least similar) to 1 (most similar). The crawler then discards articles which are insufficiently similar to the parent article.

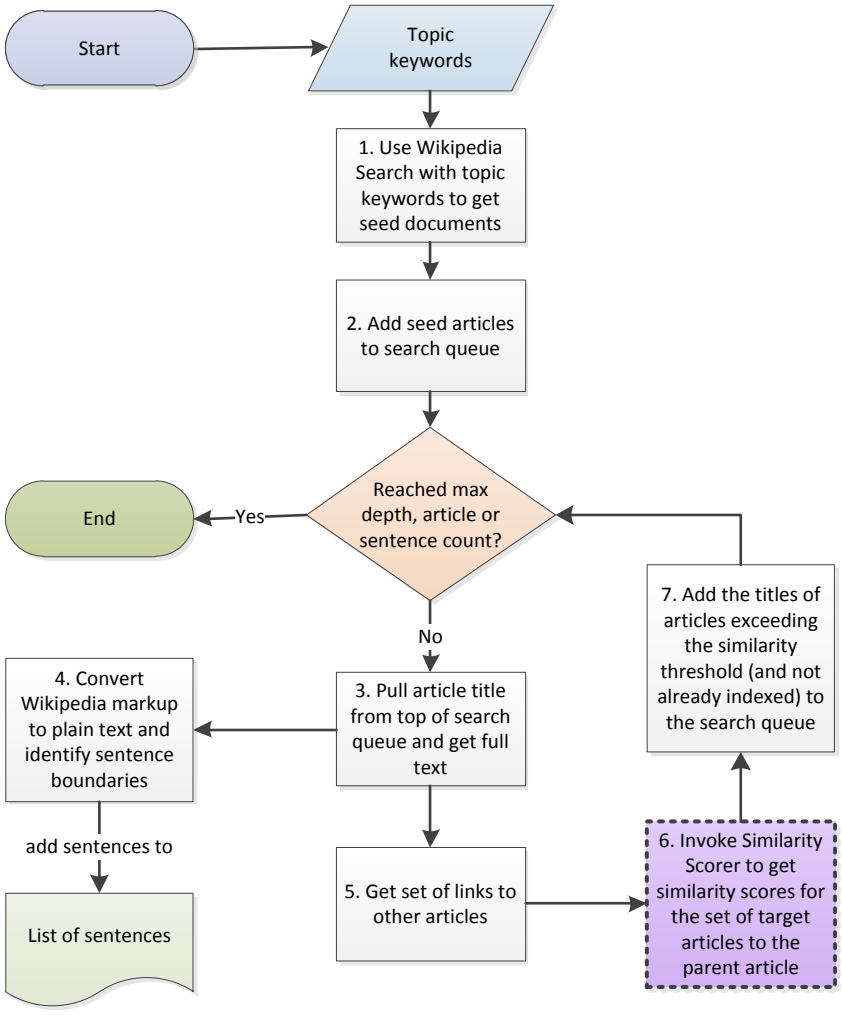


Figure 4-12: Wikipedia Crawler with Similarity Scorer

The similarity threshold applied is 0.7 + 0.025 * article_depth. Thus articles linked from depth 0 articles (those returned by the keyword search) need to have similarity >= 0.7 to be included in the index queue, whereas for links from depth 1 articles, a threshold of 0.725 is applied, and so on. This is intended to counteract topic divergence as distance from the seed articles increases.

Figure 4-13 illustrates the Similarity Crawler applied to the lecture on Lycidas as in Figure 4-8 with the same crawler constraints as in Table 4-2.

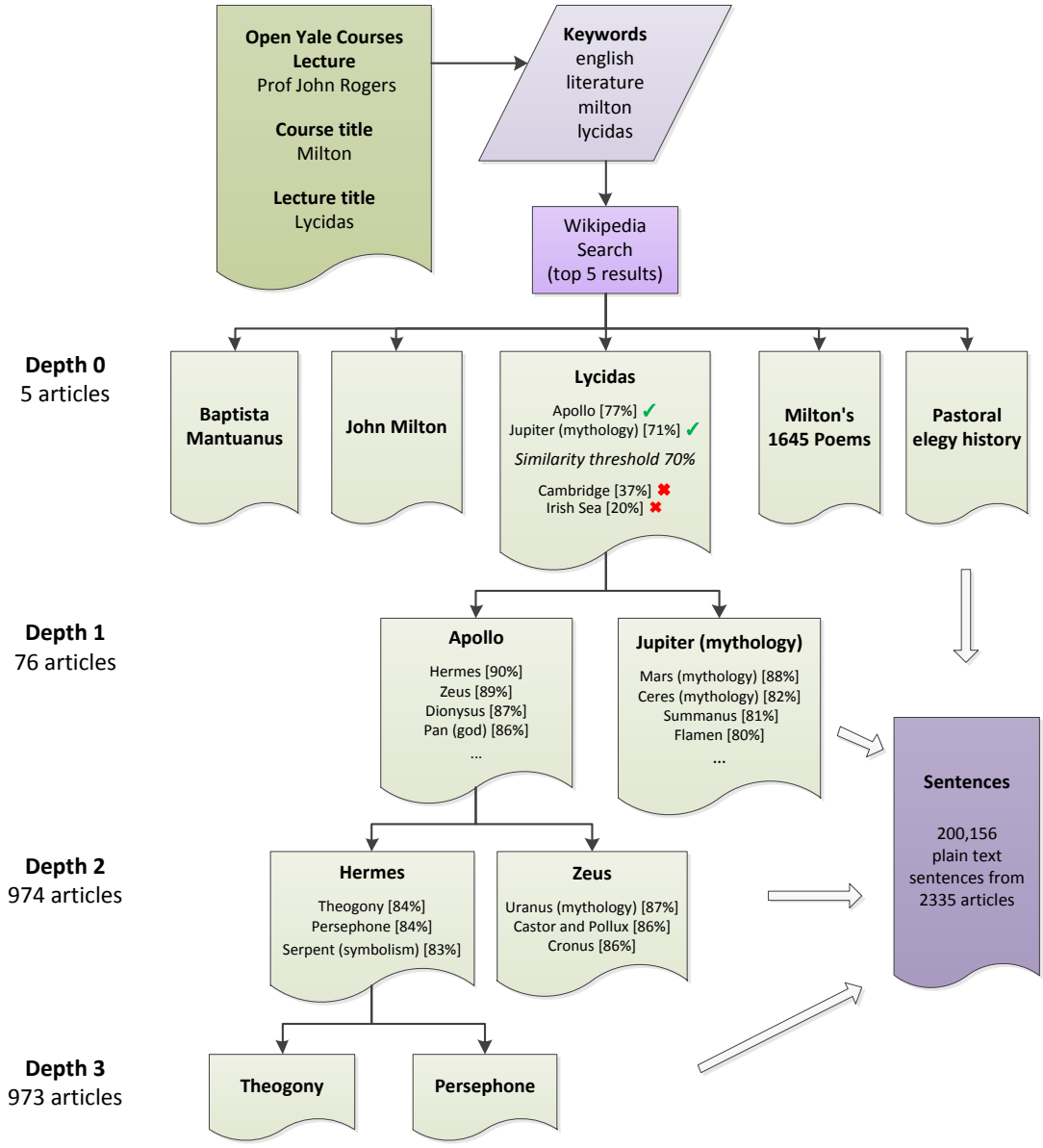


Figure 4-13: Wikipedia Crawler for lecture on Lycidas (Similarity Crawler)

In this example, the crawler outputs 200 065 sentences, having processed 2028 articles at a maximum depth of 3.

The effect of the similarity scorer is that the crawler added just under 9% of articles considered to the indexing queue (2533 out of 21229 articles considered). For example, from the top-level article "Lycidas", links to "Apollo" and "Jupiter (mythology)" are followed (article similarity scores of 0.77 and 0.71 respectively), while links such as "Cambridge" and "Irish Sea" are rejected (article similarity scores of 0.37 and 0.2).

5 Discussion and main findings

5.1 Introduction

This chapter presents the key results of the experimental work described in Chapters 3 and 4.

Limitations on the accuracy of the recognition process are noted, and the baseline recognition performance of the Sphinx 4 speech recognition engine with the reference HUB4 acoustic and language models is presented.

The behaviour and output of the naïve and similarity Wikipedia crawlers is described, and recognition performance with the HUB4, naïve and similarity language models is compared. The impact of estimated pronunciation in generated dictionaries is examined, as is the impact of extraneous words in the recognition output.

Word recognition is then further considered in terms of the goal of increasing searchability. An analysis of the recognition results of the three models ordered by word frequency is presented, showing greater divergence in recognition performance for less frequent words, supporting the introduction of a new metric: Ranked Word Correct Rate (RWCR).

Finally, the correlation of four different metrics is examined.

5.2 Limitations on accuracy

The nature of the data and methodology used introduce some limitations on the resulting accuracy figures. Five factors affecting accuracy are noted here: transcript fidelity, text conditioning, Sphinx configuration, dictionary variation, and corpus quality and drift. In general, these factors are likely to lead to a slight understatement of absolute accuracy, but do not affect relative accuracy between models in a significant way.

1. *Transcript fidelity*: the source transcripts for the Open Yale Courses are edited for readability, and thus may exclude disfluencies, repetitions or filler words such as “um”. Should these occur in the recognition output, they would be considered incorrectly as insertions.

2. *Text conditioning* is the process of converting text to a consistent, canonical form suitable for language modelling and comparing the output of recognizers with the reference transcript. For this investigation, only minimal text conditioning has been applied, which means for example literal numerals in transcripts (“58”) may not match word recognition output (“fifty-eight”).

3. A number of *Sphinx configuration* parameters affect its behaviour and the resulting output, and the optimal configuration for one recording may be less optimal for another.

4. *Dictionary variation:* English Wikipedia as a whole is considered representative of general English for this project, and is used inter alia for generating frequency-ranked word lists. However, a comparison of three dictionaries (Table 5-1) shows significant

divergence and surprisingly low overlap. The HUB4 vocabulary shares only two-thirds of its words with the top 64,000 words from English Wikipedia and the Google books ngram data set, while all three dictionaries have only 58% of words in common [90], [91].

Dictionaries	Words in common	Percentage of words in common
HUB4 / Google ngram 64K	43 173	67%
HUB4 / Wikipedia 64K	42 099	66%
Google ngram 64K / Wikipedia 64K	45 129	71%
HUB4, Google ngram 64K, Wikipedia	37 021	58%

Table 5-1: Overlap between HUB4, Wikipedia and Google dictionaries

5. *Corpus quality and drift:* Wikipedia itself is of variable quality and by design is loosely curated. Therefore it may contain many misspellings or syntactic errors in the text or markup, which affect the quality of the resulting vocabularies and language models. As it is also under constant revision, any results which depend on Wikipedia as an online resource are subject to some degree of drift as the articles evolve and static offline resources (such as a pre-computed latent semantic indexing model as described in 4.10) diverge from the related online data.

5.3 Baseline performance with HUB4

The HUB4 acoustic and language models are used as the reference models. The HUB4 language model contains 64,000 unigrams. CMUdict 0.7a is used as the baseline pronunciation dictionary. Pronunciations have been estimated for 177 words which are contained in the HUB4 language model but are not in CMUdict.

Table 5-2 presents a set of recognition statistics of the thirteen sample lectures with the HUB4 models:

- *Transcript words* is the number of words in the reference transcript.
- *OOV words* is the number of words in the reference transcript which are not contained in the dictionary.
- *Perplexity* is the calculated perplexity of the language model for the transcript without sentence markers.
- *Length* is the length of the audio recording.
- *RT ratio* is the ratio of the time spent by the recognizer (runtime) in relation to the length of the audio.
- *Transcript sentences* is the number of sentences in the reference transcript.
- *Output segments* is the number of lines output by the recognizer.
- *Output words* is the number of words in the hypothesis transcript.

Lecture	Transcript words	OOV words	Perplexity (continuous transcript)	Length (mm:ss)	RT ratio	Transcript sentences	Output segments	Output words
astr160	6,704	47	228	45:47	2.50	447	415	7,312
beng100a	7,385	54	307	52:21	2.10	336	552	8,166
beng100b	6,974	62	211	46:32	2.18	334	400	7,706
eeb122	5,795	97	331	40:55	2.82	370	484	5,935
engl220	7,350	143	535	51:51	2.63	330	556	8,293
engl291	6,201	96	379	49:27	1.61	274	503	6,596
engl300	6,701	116	274	52:49	1.60	280	299	7,992
hist116	7,902	54	309	47:44	1.84	332	492	7,797
hist202	6,643	131	252	49:56	1.73	466	427	8,031
phil176	6,603	31	475	48:52	1.99	464	509	6,676
plsc114	5,473	48	357	45:34	1.27	249	536	5,976
psyc110	7,085	70	275	48:46	1.78	467	641	7,156
rlst152	8,196	58	286	47:40	2.34	393	391	8,614
Sum	**89,012**	**1,007**				**4742**	**6205**	**96,250**
Average	**6847**	**77**	**325**	**48:20**	**2.03**			

Table 5-2: Recognition statistics with HUB4 models

Notable here is that the number of output segments (where the recognizer inserts a line break in the output based on elapsed time between utterances) exceeds the number of sentences by around 30%, and the correlation between sentences and output segments is relatively weak (0.12).

This implies that the recognizer has difficulty identifying sentence boundaries in the speech, and therefore that sentence markers in the language models will have limited value. Word Error Rate and Perplexity have thus been calculated across the whole transcript (without sentence boundaries) rather than per-sentence.

The recognizer has also output approximately 8% more words than in the original transcript, suggesting a bias towards shorter words in the Sphinx configuration (regulated partly by the wordInsertionProbability parameter) and/or language model.

Table 5-3 presents recognition accuracy for the sample lectures in terms of Word Error Rate and Word Correct Rate:

- *Edit distance* is the Levenshtein distance between the reference and hypothesis transcripts (number of insertions, deletions and substitutions required for the hypothesis to match the reference), and is used to calculate the *Word Error Rate (WER)*.
- *Words correct* is the number of words in the reference transcript which are correctly recognized in the reference transcript, and is used to calculate the *Word Correct Rate (WCR)*.

Lecture	Transcript words	Edit distance	Word Error Rate (WER)	Words correct	Word Correct Rate (WCR)
astr160	6,704	2,360	35.2%	5,185	77.3%
beng100a	7,385	2,617	35.4%	5,780	78.3%
beng100b	6,974	2,228	31.9%	5,675	81.4%
eeb122	5,795	2,312	39.9%	3,935	67.9%
engl220	7,350	3,185	43.3%	5,312	72.3%
engl291	6,201	2,015	32.5%	4,738	76.4%
engl300	6,701	3,162	47.2%	5,006	74.7%
hist116	7,902	3,131	39.6%	5,184	65.6%
hist202	6,643	4,059	61.1%	4,293	64.6%
phil176	6,603	3,230	48.9%	3,856	58.4%
plsc114	5,473	1,739	31.8%	4,355	79.6%
psyc110	7,085	2,984	42.1%	4,661	65.8%
rlst152	8,196	3,379	41.2%	5,730	69.9%
Average			**40.8%**		**71.7%**

Table 5-3: Recognition accuracy with HUB4 models (WER and WCR)

Both measures show relatively wide variation in accuracy, with WER ranging from 31.8% to 61.1% and WCR ranging from 58.4% to 81.4%. Audio factors which could account for this variation include the degree of background noise and reverberation in the recording (determined by room acoustics and microphone position), and the extent of alignment between the acoustic model and the speaker's accent.

A difficulty in examining the impact of changes in the recognition process is in understanding the extent to which acoustic or language factors dominate recognition accuracy. Nevertheless, the average WER here of around 40% is consistent with reported results from other projects and recognizers ([5], [51]).

5.4 Comparing the Wikipedia crawler behaviour and output

Table 5-4 summarises the results of executing the naïve and similarity Wikipedia crawlers with the keywords chosen for each sample lecture, with the constraints described in Table 4-2:

- *Links considered* is the number of outgoing article links which the crawler encountered.
- *Links queued* is the number of article links which the crawler added to the indexing queue.
- *Links queued %* is the proportion of links considered which are added to the indexing queue. For the Naïve Crawler, all links are followed, whereas for the Similarity Crawler, only links to articles which meet the similarity threshold are adding to the indexing queue.

- *Docs indexed* is the number of articles converted to plain text and added to the output corpus.
- *Total sentences* is the number of plain text sentences added to the output corpus.
- *Total Words* is the number of words in the output corpus.

Lecture	Links considered	Links queued	Links queued %	Docs indexed	Total sentences	Total Words
Naïve Crawler						
astr160	2,592	2592	100%	2462	200,113	4,769,510
beng100a	2,909	2909	100%	1852	200,123	4,458,388
beng100b	2,589	2589	100%	2500	115,229	2,689,101
eeb122	2,583	2583	100%	2500	158,760	3,527,469
engl220	2,609	2609	100%	2335	200,156	4,752,762
engl291	2,606	2606	100%	1752	200,067	4,782,250
engl300	2,624	2624	100%	2291	200,008	4,738,475
hist116	2,634	2634	100%	2500	178,178	4,325,914
hist202	2,548	2548	100%	1845	200,013	4,836,736
phil176	2,662	2662	100%	2253	200,095	4,724,906
plsc114	2,528	2528	100%	2162	200,339	4,695,799
psyc110	2,540	2540	100%	1739	200,200	4,712,681
rlst152	2,762	2762	100%	2155	200,228	4,829,361
Average	**2,630**	**2,630**		**2,180**	**188,731**	**4,449,489**
Similarity Crawler						
astr160	10,282	2619	25%	2500	180,300	4,372,655
beng100a	14,850	2515	17%	2500	184,621	4,174,711
beng100b	23,869	2557	11%	2500	163,045	3,852,979
eeb122	13,876	2539	18%	2500	135,829	2,881,927
engl220	21,299	2533	12%	2028	200,065	4,851,670
engl291	31,750	2535	8%	2292	200,011	4,742,750
engl300	14,765	2501	17%	2083	200,358	4,766,094
hist116	26,782	2519	9%	1490	200,087	4,866,393
hist202	19,715	2514	13%	1656	200,044	4,846,217
phil176	11,492	2525	22%	2103	200,127	4,769,079
plsc114	19,033	2500	13%	2047	200,088	4,788,145
psyc110	12,892	2576	20%	2333	200,094	4,657,753
rlst152	12,755	2508	20%	2197	200,010	5,033,929
Average	**17,951**	**2,534**	**14%**	**2,171**	**189,591**	**4,508,023**

Table 5-4: Wikipedia Crawler Statistics

The Similarity Crawler followed between 8% and 25% of links considered (highlighted in purple above). Applying the similarity threshold to be more selective about which article links to follow means that the Similarity Crawler considered many more articles

in order to reach the same number of indexed articles (or output sentences) as the Naïve Crawler, and thus the total links considered is higher.

The average depth of articles harvested by the Similarity Crawler is thus also greater (where depth is the link-distance from the seed articles), with many more articles at depth 2 and some at depth 3 in the Similarity Crawler's output. Figure 5-1 shows the percentage of articles indexed at each depth for each lecture.

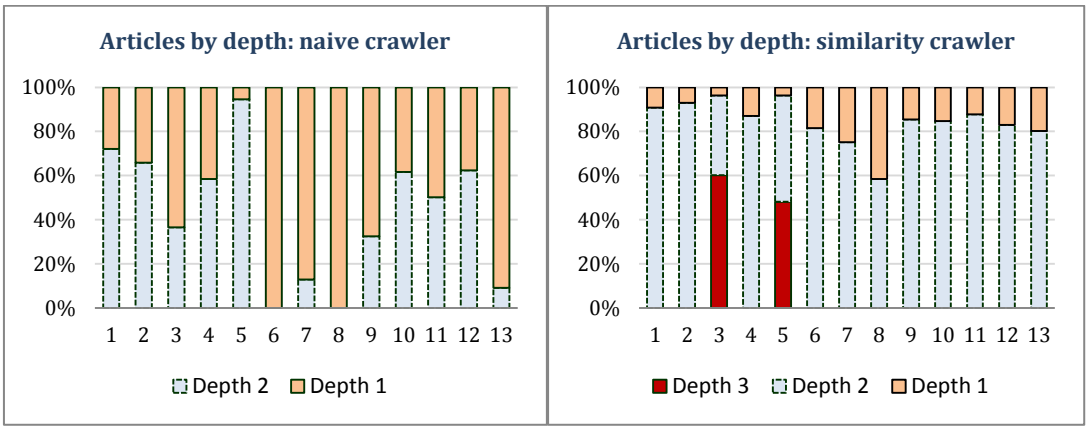


Figure 5-1: Percentage of Wikipedia articles by depth for Naïve and Similarity Crawlers

The variation across lectures in the percentage of links queued (and thus total number of links considered and depth reached in order to generate an equivalent number of output sentences) suggests that topic coverage and connectedness is uneven in Wikipedia.

As the similarity metric is a measure of distance, topic coverage in Wikipedia could be expressed in terms of density. Where seed articles have a large number of links to strongly-related other articles (reflecting both semantic relevance and connectedness), topic density could be regarded as high, whereas for topics where similarity scores are lower and there are fewer links, the topic has sparser coverage.

While the concept of variations in topic density is not explored further here, one could perhaps expect that more robust language models would be created from topics with higher density coverage in Wikipedia than those for which coverage is sparse.

Table 5-5 shows the impact of the two crawler strategies on the resulting vocabulary:

- *Articles in both naïve and similarity* is the overlap in articles indexed by both crawlers, calculated as the number of articles in common as a percentage of articles indexed by the Similarity Crawler.
- *Vocabulary in both naïve and similarity* is the overlap in vocabulary (unique words) in the language models derived from the crawler outputs, calculated as the number of words in common (after word frequency cut-offs are applied, as described in Step 3 in 4.5), as a percentage of the target language model size (64,000 unigrams).
- *Words not in Wikipedia 64K* is the number of words for each language model which do not occur in the top 64,000 words from a frequency-ranked English Wikipedia dictionary.

Lecture	Articles in both naïve and similarity	Vocabulary in both naïve and similarity	Words not in Wikipedia 64K: naïve crawler	Words not in Wikipedia 64K: similarity crawler
astr160	33%	96%	4,854	5,192
beng100a	9%	89%	3,082	7,550
beng100b	11%	94%	2,241	4,251
eeb122	30%	94%	6,473	7,505
engl220	13%	90%	5,879	7,963
engl291	18%	92%	3,416	5,651
engl300	29%	94%	5,352	6,244
hist116	15%	93%	3,378	5,312
hist202	19%	92%	5,066	6,119
phil176	36%	94%	5,720	6,774
plsc114	23%	92%	5,397	7,143
psyc110	10%	91%	3,536	6,202
rlst152	24%	93%	6,115	6,999
Average	**21%**	**93%**	**4,655**	**6,377**

Table 5-5: Articles and word comparison between Naïve and Similarity Crawler output

While the Naïve and Similarity Crawlers select very different sets of articles with only 21% in common on average, the vocabulary of the resulting language models is more similar, with an average overlap of 93%.

However, in all cases the vocabulary generated from the articles indexed by the Similarity Crawler is more specialized, i.e. it contains more words which do not occur in the top 64,000 words from a frequency-ranked Wikipedia dictionary (on average, 6377 compared to 4655). The Similarity Crawler thus appears to be more effective at selecting articles with a topic specialization as reflected in the use of a specialized vocabulary.

While the variation in vocabulary appears relatively modest (a difference of 1722 words on average), the largest number of unique words in any of the sample transcripts is only 1591, so a relatively small difference in language model vocabulary can be significant for recognition accuracy.

Table 5-6 compares the average sizes of the custom language models generated for each lecture and crawler strategy to the reference HUB4 language model:

- *Unigrams* is the number of unique words in the language model.
- *Bigrams* is the number of two-words combinations in the language model.
- *Trigrams* is the number of three-word combinations in the language model.

- *Total* size is the number of entries in the language model (unigrams, bigrams and trigrams).

Model	Unigrams	Bigrams	Trigrams	Total size
HUB4 LM	64,001	9,382,014	13,459,879	22,905,894
Average of Wikipedia Naïve LMs	64,219	9,139,981	32,600,005	41,804,205
Average of Wikipedia Similarity LMs	64,219	9,126,390	32,605,170	41,795,779

Table 5-6: Average sizes of the HUB4 and Wikipedia Language Models

The custom models are sized to have the same vocabulary (number of unigrams) as the HUB4 model for comparison purposes, and have a similar number of bigrams but approximately twice as many trigrams. While the design of the HUB4 model is not documented in detail, this difference is most likely explained by the application of a frequency cut-off for HUB4 trigrams, which has not been applied to the custom models.

5.5 Recognition performance of Naïve and Similarity language models

To investigate whether the Similarity Crawler produces better language models than the Naïve Crawler, the recognition performance of the language models derived from the respective Wikipedia crawlers is compared across four metrics: out-of-vocabulary words, perplexity, WER and WCR.

In Table 5-7:

- *Sum of unique OOV words* represents the extent to which the language model vocabulary is aligned with that of the transcript: the sum of the number of unique words for each lecture which were not included in the generated language model (lower is better).
- *Average perplexity* is a theoretical measure of the alignment between the language model and the reference transcript (lower is better).
- *Average WER* is the average Word Error Rate of the hypothesis transcripts (lower is better).
- *Average WCR* is the average Word Correct Rate of the hypothesis transcripts (higher is better).

Language models	Sum of unique OOV words	Average Perplexity	Average WER	Average WCR
Naïve LM	791	297	41.9%	68.0%
Similarity LM	735	248	41.6%	68.4%

Table 5-7: Comparison of recognition performance for Naïve and Similarity LMs

The Similarity language models thus outperform the naïve language models across all metrics, although in some cases by a relatively small amount. Vocabulary coverage is improved, perplexity is lower, WER is better though by only 0.3%, and WCR is better by 0.4%.

To investigate whether the Similarity Crawler's language models are uniformly better than the Naïve Crawler's models, Table 5-8 shows the performance increase (positive) or decrease (negative) between the performance of the Similarity and Naïve Crawler language models by individual lecture across the same metrics:

Lecture	Decrease in unique out-of-vocabulary words	Decrease in perplexity	Decrease in WER (absolute %)	Increase in WCR (absolute %)
astr160	3	7	0.7%	0.7%
beng100a	21	314	1.7%	1.5%
beng100b	-8	167	-1.2%	-1.0%
eeb122	-1	11	0.5%	0.7%
engl220	24	20	1.5%	0.6%
engl291	3	18	-0.1%	0.1%
engl300	9	15	-0.4%	-0.4%
hist116	-1	2	-0.2%	-0.1%
hist202	-12	-2	-0.1%	0.0%
phil176	0	17	0.0%	0.3%
plsc114	4	17	1.1%	1.0%
psyc110	11	44	0.9%	0.7%
rlst152	3	13	0.5%	0.5%
Average	**4**	**49**	**0.4%**	**0.4%**

Table 5-8: Relative performance of naïve and similarity LMs per lecture

For the seven lectures highlighted above (shaded green), the similarity LM outperforms the naïve LM across all metrics. For the other six, the performance is marginally worse, or better in some metrics and worse for others.

The application of the similarity filter for the crawler therefore does provide a net benefit, though somewhat unevenly.

5.6 Recognition performance of HUB4 and Similarity language models

To investigate whether the Similarity Crawler language models outperform the reference HUB4 language model, the recognition performance of the Similarity language models is compared to that of the reference HUB4 language model across four metrics in Table 5-9:

Language models	Sum of unique out-of-vocabulary words	Average Perplexity	Average WER	Average WCR
HUB4	1,007	324	40.8%	71.7%
Similarity	735	248	41.6%	68.4%

Table 5-9: Comparison of recognition performance for HUB4 and Similarity LMs

Table 5-10 shows the performance increase (positive, highlighted green) or decrease (negative, highlighted red) by individual lecture across the same metrics:

Lecture	Decrease in unique out-of-vocabulary words	Decrease in perplexity	Decrease in WER (absolute %)	Increase in WCR (absolute %)
astr160	27	53	-0.6%	-3.2%
beng100a	36	106	2.2%	-0.9%
beng100b	-1	-33	-4.7%	-6.4%
eeb122	37	87	-1.0%	-3.0%
engl220	15	191	1.7%	-1.1%
engl291	-1	60	-3.3%	-4.2%
engl300	48	28	-0.3%	-3.6%
hist116	-10	35	-2.8%	-4.5%
hist202	34	18	0.8%	-3.7%
phil176	-1	189	-0.7%	-3.5%
plsc114	22	148	1.5%	-1.3%
psyc110	28	25	-2.4%	-5.1%
rlst152	38	87	-0.2%	-2.9%
Avg	**21**	**76**	**-0.8%**	**-3.3%**

Table 5-10: Relative performance of HUB4 and Similarity LMs per lecture

One lecture (beng100b) is worse across all metrics, whereas the other 12 show improved perplexity, but mixed or worse performance in vocabulary, WER and WCR.

Thus on average the Similarity language models outperform HUB4 in the two language-related metrics (out-of-vocabulary words and perplexity), but recognition performance for the Similarity LMs reflected in WER and WCR is actually worse, with an increase in WER of 0.8% and decrease in WCR of 3.3%.

5.7 Effect of estimated pronunciation

As an aim of the Wikipedia crawler is to introduce specialist vocabulary into the topic-adapted language models, it is likely that a number of such words will not occur in the relatively small CMU pronouncing dictionary (around 123,000 words). Pronunciations for such words are therefore estimated, in this project through the phonetisaurus grapheme-to-phoneme tool using a model trained from CMUdict [83], [92].

As these estimations are extrapolations of implicit rules in CMUdict, they may be inaccurate for unusual or foreign vocabulary and thus produce poor recognition results. For example, Table 5-11 shows word recognition results for the word "Lycidas" from the lecture on Milton with estimated (machine-generated) and manual (human-generated) pronunciations:

Pronunciation	Word recognition count (Wikipedia Similarity LM)
L AY S AH D AH Z	0
L IH S IY D AH S	7 / 47

Table 5-11: Recognition of Lycidas with variant pronunciations

With the estimated pronunciation, the word is not recognized at all (although occurring 47 times). With a manual pronunciation closer to how the word is spoken, the recognition rate improves to 7 instances out of 47.

To investigate whether pronunciations estimated by phonetisaurus lead to lower word recognition rates than those for words with pronunciations in the CMU Dictionary, the recognition rates of the respective word types are compared in Table 5-12:

- *Words from CMUdict* is the number of words in the transcript which have pronunciations in the CMU Pronouncing Dictionary (CMUdict)
- *CMUdict words recognized at least once* is words found in CMUdict which are recognized correctly at least once (i.e. occur in the output transcript)
- *CMUdict word recognition rate* is the word correct rate for words found in CMUdict (number of words recognized correctly as a percentage of total words)
- *Est. words* is the number of words in the transcript without pronunciations in CMUdict, and thus for which pronunciations have been estimated by phonetisaurus.
- *Est. words recognized at least once* is words with estimated pronunciations which are recognized correctly at least once
- *Est. word recognition rate* is the word correct rate for words with estimated pronunciation

	Words from CMUdict	CMUdict words recognized at least once	CMUdict word recognition rate	Est. words	Est. words recognized at least once	Est. word recognition rate
astr160	975	85%	76%	28	54%	69%
beng100a	1050	89%	80%	30	60%	45%
beng100b	1078	87%	78%	11	45%	47%
eeb122	1107	77%	67%	35	51%	51%
engl220	1503	82%	75%	43	47%	36%
engl291	1437	80%	75%	27	37%	30%
engl300	1283	82%	75%	51	43%	26%
hist116	1417	79%	64%	7	43%	27%
hist202	1406	71%	64%	48	15%	15%
phil176	787	76%	57%	14	50%	52%
plsc114	1101	87%	81%	25	40%	36%
psyc110	1231	79%	62%	29	62%	58%
rlst152	1312	81%	69%	40	63%	59%
Total	**15687**			**388**		
Average	**1207**	**81%**	**71%**	**30**	**46%**	**39%**

Table 5-12: Recognition rate of words with estimated pronunciation

For CMUdict pronunciations, 81% of all words are recognized at least once, whereas only 46% of words with estimated pronunciations are ever recognized correctly. Estimating pronunciation is therefore only partially effective.

If the recognition rate of estimated-pronunciation words matched those with CMUdict pronunciations (i.e. increased from 39% to 71%), an additional 6 unique words or 22 additional words in total could be recognized correctly on average per lecture. While the absolute number of estimated-pronunciation words is quite small (an average of 30 per lecture), these words are likely to be significant for searchability.

5.8 Introduction of extraneous words

Recognition failures not only lead to the omission of a correct word in the hypothesis transcript, but they also introduce an incorrect word. A class of incorrect words are those which do not occur at all in the reference transcript, identified here as extraneous words. Considered in terms of the information retrieval measures recall and precision, the introduction of extraneous words in a transcript lowers the precision of search results by increasing false positives, i.e. matches to words in the transcript which should not be there.

To investigate the extent to which the Similarity LMs introduce extraneous words, the number of extraneous words introduced by the HUB4 and Similarity LMs respectively is shown in Table 5-13:

Lecture	Extraneous words: HUB4 LM	Extraneous words: Similarity LM	% increase in extraneous words
astr160	422	459	9%
beng100a	486	504	4%
beng100b	415	582	40%
eeb122	595	666	12%
engl220	671	749	12%
engl291	586	733	25%
engl300	624	756	21%
hist116	625	829	33%
hist202	806	967	20%
phil176	588	759	29%
plsc114	381	401	5%
psyc110	510	635	25%
rlst152	617	715	16%
Total	**7,326**	**8,755**	**20%**

Table 5-13: Extraneous words introduced by the HUB4 and Similarity LMs

On average, the Similarity LM introduces 20% more extraneous words than the HUB4 LM, although as the number of extraneous words is strongly correlated with Word Error Rate (with a correlation of 0.84), this effect is worst for lectures where the Similarity LM led to an increase in overall WER.

An example of the introduction of extraneous words in the recognition outputs for the lecture on Milton (engl220) is shown in Table 5-14, which lists extraneous words which occur three or more times in the transcripts produced by the HUB4 and Similarity LMs respectively.

Of note for this lecture is that while the absolute number of extraneous words increased by 12%, the distribution for these words in the results from the Similarity LM contains a longer tail than those from the HUB4 LM, as the Similarity results show slightly fewer unique extraneous words with frequency 3 or more.

These words are additionally slightly less common than those introduced by HUB4, with a median rank of 3582 vs 1856, again showing that the Wikipedia-derived language models include a more specialized vocabulary than that of HUB4.

A number of words listed here are artefacts of variant spellings (for example “Masque” vs “Mask”) or text conditioning differences (“sixteen” vs “16”) while other content terms such as “Odyssey”, “Iris”, “Hera”, and “Kant” are probable search terms and could cause the transcript to incorrectly appear in search results.

Extraneous words introduced by both language models (freq>=3)			Extraneous words introduced only by HUB4 LM (freq>=3)			Extraneous words introduced only by Similarity LM (freq>=3)		
Word	**Dict rank**	**Doc freq**	**Word**	**Dict rank**	**Doc freq**	**Word**	**Dict rank**	**Doc freq**
ODYSSEY	11498	13	DOCTOR	1448	8	USED	52	6
LISTS	1463	11	HAPPEN	2761	5	HERA	27997	5
SOLICITOUS	185170	8	BOB	1624	5	SONNETS	22670	4
THIRTY	4174	5	EIGHT	952	5	LISTED	745	4
SIXTEEN	5280	4	HILTON'S	68444	4	WITHIN	212	4
SELF	3061	4	EIGHTY	14238	4	ABHORRENCE	109012	3
COLUMN	2913	3	OLDER	862	4	KANT'S	40806	3
BILL	920	3	REPORT	590	4	MASQUE	26785	3
GOT	679	3	ORIOLE	30973	3	TANGLED	25182	3
HOME	207	3	DAY'S	11684	3	IRIS	10794	3
			ANALOGY	10026	3	ABBOT	8171	3
			LISTENERS	8548	3	PALM	4727	3
			POLL	4396	3	WRIGHT	3582	3
			BOMB	2903	3	HANDLE	3581	3
			SCENES	2611	3	REPORT	590	3
			YOUNGER	1856	3	WHITE	326	3
			CRIME	1753	3	AIR	276	3
			GUY	1724	3	EDITS	271	3
			BAD	943	3	MEMBER	213	3
			COURT	366	3			
			FIVE	311	3			
Median rank:	**2987**		**Median rank:**	**1856**		**Median rank:**	**3582**	

Table 5-14: Comparison of extraneous words in recognition output of Lycidas lecture

5.9 Relation to searchability

The results in 5.6 show that the topic-adapted language models have a net negative effect on WER and WCR. The resulting transcripts are therefore likely to be less readable, and in information retrieval terms, have lower precision with the introduction of more extraneous words.

However, in relation to the goal of improving searchability, not all words are created equal: users are more likely to use less common words as search terms. Do the topic-adapted language models therefore lead to more searchable transcripts, or defined in information retrieval terms, provide better recall?

To gain insight into possible qualitative differences in recognition performance between language models, differential word recognition rates are examined in the Milton lecture. Table 5-15 shows the top and bottom groups of words for this lecture where the recognition rate diverges most between the Similarity and HUB4 language models:

- *Dictionary Rank* is the word's position in a frequency-ranked Wikipedia English dictionary

- *Word in CMU Dict* indicates whether the word is found in the CMU Pronouncing Dictionary (if not, pronunciation has been estimated)
- *Word in HUB4 Dict* indicates whether the word occurs in the vocabulary of the HUB4 language model
- *Recognized with HUB4 LM* is the number of times that the word was correctly recognized using the HUB4 language model
- *Recognized with Wikipedia Similarity LM* is the number of times that the word was correctly recognized using the Similarity language model
- *Word Freq in Doc* is the number of times that the word occurs in the transcript
- *Recognition increase or decrease* is the difference in word recognition count between the Similarity and HUB4 language models.

Word	Dictionary Rank	Word in CMU Dict	Word in HUB4 Dict	Recognized with HUB4 LM	Recognized with Wikipedia Similarity LM	Word Freq in Doc	Recognition increase or decrease
Top 15 words where recognition rate with Wikipedia Similarity LM exceeds HUB4 LM							
MILTON'S	41755	0	0	0	29	35	+29
POEM	2999	1	1	5	17	44	+12
GOD	865	1	1	12	21	22	+9
HEAVEN	4191	1	1	0	9	14	+9
KING	291	1	1	6	14	18	+8
POET	1919	1	1	20	27	34	+7
ELDER	3913	1	1	8	14	15	+6
HIS	9	1	1	43	49	64	+6
ORPHEUS	21669	1	1	8	14	17	+6
COMUS	91192	0	0	0	5	10	+5
EDWARD	1150	1	1	5	10	16	+5
MILTON	6092	1	1	65	70	74	+5
MUSE	10609	1	1	1	6	10	+5
THEOCRITUS	111859	0	0	0	5	5	+5
DIODATI	187129	1	0	0	4	6	+4
Bottom 15 words where recognition rate with Wikipedia Similarity LM is lower than HUB4 LM							
THERE'S	1009	1	1	3	0	6	-3
ABLE	520	1	1	10	6	11	-4
ALL	28	1	1	40	36	44	-4
CAN'T	768	1	1	4	0	7	-4
DEATH	272	1	1	21	17	24	-4
I'M	273	1	1	5	1	9	-4
NOW	98	1	1	16	12	22	-4
THAN	57	1	1	9	5	11	-4
FOR	4	1	1	49	44	54	-5
HERE	154	1	1	12	7	32	-5
LOOK	563	1	1	12	7	15	-5
THIS	8	1	1	114	108	144	-6
AND	2	1	1	127	109	175	-18
THAT	5	1	1	136	118	161	-18
IT'S	164	1	1	39	19	48	-20

Table 5-15: Word recognition comparison for Lycidas lecture with HUB4 and Similarity LMs

It is notable here that words whose recognition rates are increased by the Similarity LM are generally less common (as reflected by dictionary frequency rank) and longer words, whereas those for which the recognition rate has decreased are more common, shorter words.

This example suggests that the adapted LM may be improving recognition of specialist vocabulary at the expense of overall recognition: better recall, but poorer readability.

To explore this hypothesis systematically across all lectures, word recognition performance is examined across four word rank frequency groupings, using a 1.5 million word-frequency dictionary derived from English Wikipedia (words of 3 more or characters ordered from most to least frequent):

- Words with dictionary frequency rank of 1 to 1,000
- Words with dictionary frequency rank from 1,000 to 10,000
- Words with dictionary frequency rank from 10,000 to 100,000
- Words with dictionary frequency rank above 100,000

Figure 5-2 shows the distribution of all words in the set of transcripts across these rank frequency groups, with around 90% of words falling into the top two categories and the remaining 10% in the bottom two:

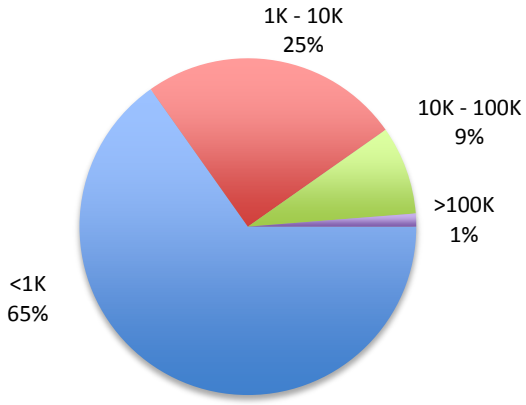


Figure 5-2: Transcript word distribution by frequency rank group

To illustrate the analysis of recognition performance by word frequency rank, Table 5-16 shows the transcript and recognition hypotheses with the HUB4 and Similarity language models of the opening sentences of the Milton lecture. Words with frequency rank of 10,000 and below are highlighted.

Transcript (before conditioning)	The best way, I think, to introduce the central issues of this wonderful poem, *Lycidas,* is to return to Milton's *Comus*. So yet once more -- and I promise this will be one of the last times that we look back at Milton's mask -- but yet once more, let's look at *Comus.* Now you will remember that the mask *Comus* was everywhere concerned with questions of the power of -- well, the strangely intertwined questions of the power of chastity on the one hand and the power of poetry on the other.
HUB4 language model	the best way to buy thank to introduce the central issues of of it's a wonderful column was so this is is to return set to milkens common so we can once more i promise this will be one of the last times that we look back at hilton's masked but what's yet once more let's look at our comments making remember now the mass comments was everywhere concerned with questions of the power of well because strangely intertwined questions of the power of chassis on the one hand the power of poetry on the other
Wikipedia similarity language model	the best ways i think to introduce the central issues that of this wonderful paul was a bus is used to return set to milton's comus odette whence more i promise this will be while the last times that we look back at milton's masque that once yet once more lives lookout a comments making remember that the masque comus was everywhere concerned questions of the power of boudica strangely intertwined questions of the power of chassis on the one hand the power of poetry on the other

Table 5-16: Transcription of opening sentences of Lycidas lecture

In this short example, the Similarity LM has recognized 6 of the 9 highlighted words correctly, compared to only 2 of 9 for the HUB4 LM. Of these “Milton” and “Comus” are likely search terms, whereas “strangely” and “intertwined” are memorable but are less likely to be used for discovery purposes.

Figure 5-3 presents the Word Correct Rate for words in each frequency rank group for each lecture by language model. For example, the WCR for 1K – 10K is given by calculating the word recognition rate for all words ranked from 1,000 to 10,000 in the English Wikipedia frequency-ranked dictionary.

The HUB4 language model (shown in blue) outperforms the Naïve (red) and Similarity (green) in all cases for words under rank 1000; from 1K to 10K results are similar, whereas for 10K – 100K and above, the Naïve and Similarity models outperform HUB4 in most cases.

This effect is notable in the majority of lectures (for example astr160, engl330, and rlst152), while in a small number (for example beng100b, engl291) the adapted models provide no clear improvement.

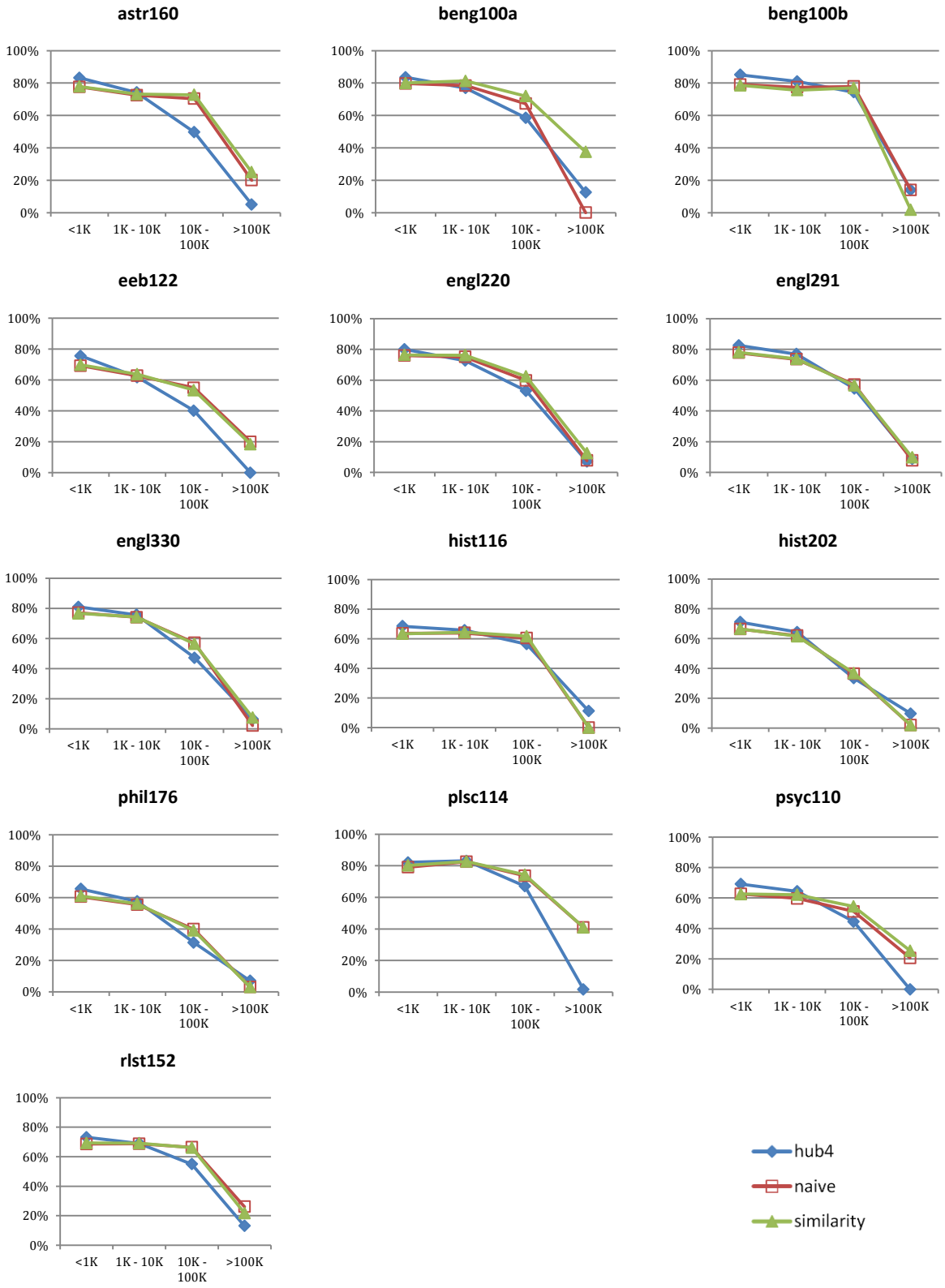


Figure 5-3: Partial Word Correct Rate by word frequency rank groups

5.10 Ranked Word Correct Rate Metric

Examining recognition accuracy by word frequency rank is therefore helpful in providing a better characterisation of the performance of topic-adapted language models in recognizing less common words. To simplify such analysis, a single metric is proposed: Ranked Word Correct Rate (RWCR-n).

RWCR-n is defined as the Word Correct Rate for all words in the document which are not found in the first n words in a given general English word dictionary (d) with words ranked from most to least frequent. At n=0, RWCR is identical to WCR and may diverge as n increases, thus:

$$\lim_{n \to 0} RWCR(n, d) = WCR$$

To illustrate the effect of a value for n of 10,000, Figure 5-4 shows the cumulative Word Correct Rate for three example lectures by language model where there are significant, moderate and negligible differences in recognition performance between the models.

In these graphs, the transcript words are arranged in inverse frequency rank order from least frequent (x=1) to most frequent (right-most word). The dotted-red line indicates the position of the word with frequency rank 10,000, and thus the separation between language models given by the metric RWCR-10K.

While the cut-off value of 10,000 is in some senses arbitrary, the examples suggest that this metric provides reasonable insight into differences in recognition performance, while still being based on a sufficient proportion of total words so as not to be too idiosyncratic.

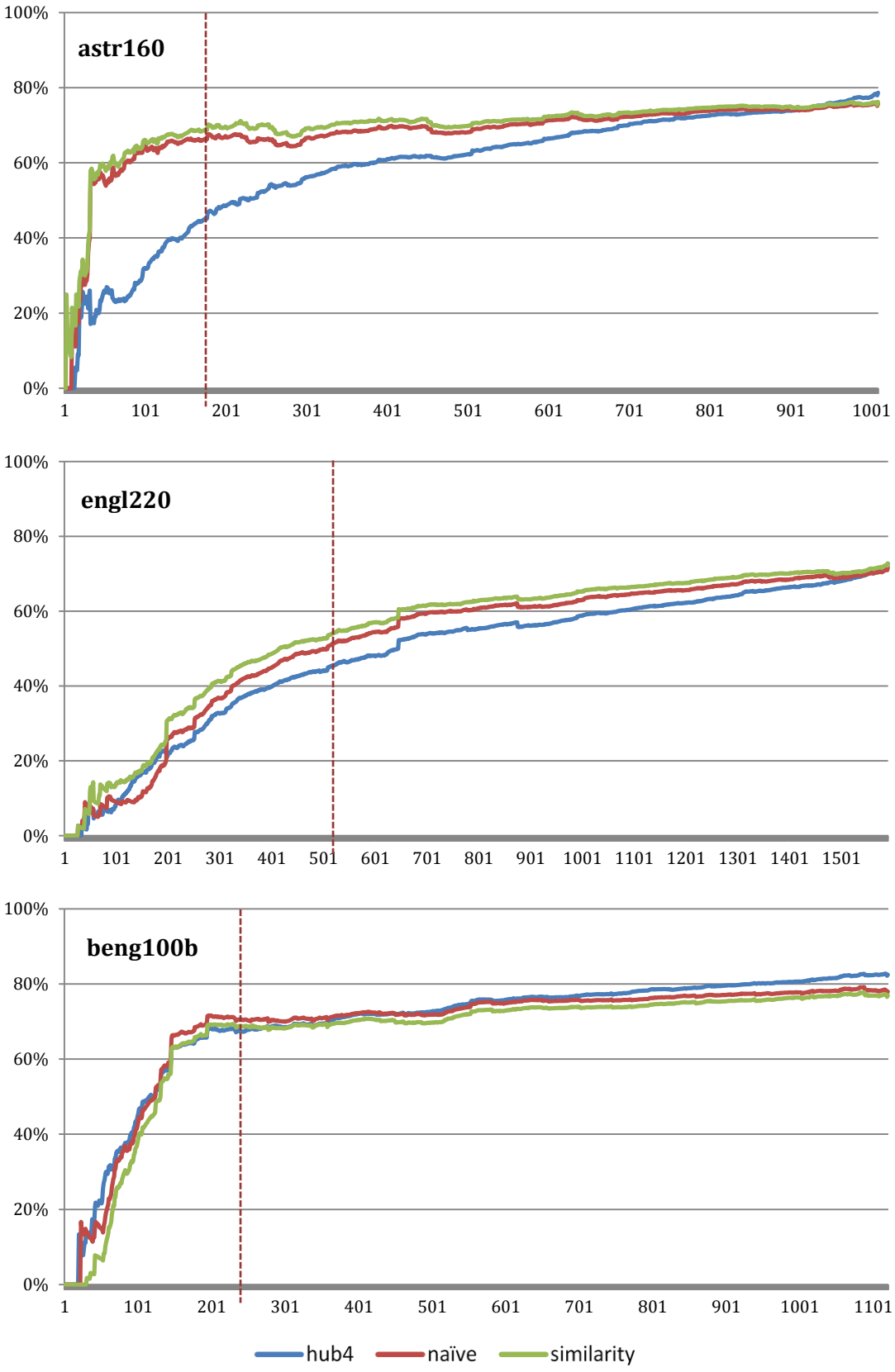
Table 5-17 shows the performance of the HUB4 and Similarity LMs across all lectures for five metrics: the four metrics shown in 5.6 (Table 5-9), and the RWCR-10K metrics:

Language models	Linguistic metrics		Recognition metrics		
	Sum of unique OOV words	Average Perplexity	Average WER	Average WCR	Average RWCR-10K
HUB4	1,007	324	40.8%	71.7%	46.0%
Similarity	735	248	41.6%	68.4%	54.9%
Difference	**272**	**76**	**0.8%**	**3.3%**	**9.0%**

Table 5-17: Comparison of recognition performance for HUB4 and Similarity LMs

RWCR-10K improves by 9% from the HUB4 to Similarity LM, even though WER worsens on average by 0.8% and overall WCR worsens by 3.3%.

Using the RWCR-10K metric, it appears therefore the topic-adapted language models are successful in improving recognition of less common words, although they do so at the expense of recognition of more common words and thus overall accuracy.



dotted line = dictionary frequency rank of 10,000

Figure 5-4: Cumulative Word Correct Rate by inverse word frequency rank

To investigate whether the RWCR-10K metric shows a uniform improvement from HUB4 for the Wikipedia language models, the recognition performance of the three language models using the RWCR-10K metric for individual lectures is presented in Table 5-18, with the best-performing LM highlighted green:

Lecture	RWCR-10K HUB4 LM	RWCR-10K Naïve LM	RWCR-10K Similarity LM
astr160	47.1	67.4	70.0
beng100a	55.2	62.4	69.4
beng100b	68.0	71.1	69.1
eeb122	35.2	50.7	49.2
engl220	44.2	49.8	52.7
engl291	46.2	48.2	48.0
engl300	40.2	47.6	48.0
hist116	53.5	56.6	57.8
hist202	31.0	32.9	33.3
phil176	27.2	33.6	32.9
plsc114	59.0	69.6	70.2
psyc110	38.4	46.9	50.4
rlst152	52.4	63.9	63.4
Average	**46.0**	**53.9**	**54.9**

Table 5-18: Ranked Word Correct Rate (10K) by lecture and language model

In all cases, both the Similarity and Naïve LMs score higher than the HUB4 LM, with the Similarity LM being the best-performing model in 8 out of 13 cases.

5.11 Correlation of metrics

To investigate how the different metrics relate to each other, the relative performance of the Similarity LM to the HUB4 LM is shown in Table 5-19 across five metrics (OOV words, Perplexity, WER, WCR and RWCR-10K), with positive effects highlighted in green.

RWCR-10K exhibits a greater range than either WER or WCR, and appears the most helpful metric in characterising the performance of the topic-adjusted language models in relation to searchability.

A decrease in out-of-vocabulary words is a reasonable predictor of RWCR performance: where the number of OOV words actually increases, improvement in RWCR is lowest. One exception to this is hist202, which shows improvement in OOV but still a relatively low RWCR improvement (2.3%), possibly on account of the high initial word error rate for that recording (61.1%).

Lecture	Absolute HUB4 WER	Decrease in unique OOV words	Decrease in perplexity	Decrease in WER	Increase in WCR	Increase in RWCR-10K
astr160	35.2%	27	53	-0.6%	-3.2%	22.9%
beng100a	35.4%	36	106	2.2%	-0.9%	14.2%
beng100b	31.9%	-1	-33	-4.7%	-6.4%	1.1%
eeb122	39.9%	37	87	-1.0%	-3.0%	13.9%
engl220	43.3%	15	191	1.7%	-1.1%	8.6%
engl291	32.5%	-1	60	-3.3%	-4.2%	1.8%
engl300	47.2%	48	28	-0.3%	-3.6%	7.8%
hist116	39.6%	-10	35	-2.8%	-4.5%	4.3%
hist202	61.1%	34	18	0.8%	-3.7%	2.3%
phil176	48.9%	-1	189	-0.7%	-3.5%	5.7%
plsc114	31.8%	22	148	1.5%	-1.3%	11.2%
psyc110	42.1%	28	25	-2.4%	-5.1%	12.1%
rlst152	41.2%	38	87	-0.2%	-2.9%	11.1%
Average	**40.8%**	**21**	**76**	**-0.8%**	**3.3%**	**9.0%**

Table 5-19: Relative performance of Similarity LM to HUB4 LM by lecture across four metrics

Finally, to investigate the strength of the relationship between the metrics Perplexity, WER, WCR and RWCR-10K, the statistical correlation is calculated using the set of results for the 13 lectures each with the HUB4, Naïve and Similarity language models (i.e. a set of 39 data points for each metrics), as shown in Table 5-20:

Correlation	Perplexity	WER	WCR	RWCR-10K
Perplexity		0.05	-0.12	-0.27
WER	0.05		-0.73	-0.75
WCR	-0.12	-0.73		0.63
RWCR-10K	-0.27	-0.75	0.63	

Table 5-20: Correlation of WER, WCR, Perplexity and RWCR-10K metrics

Perplexity shows weak correlation with all other metrics, which is somewhat counter-intuitive but suggests that optimizing language models for low perplexity is not necessarily a good strategy for improving recognition performance.

WER, WCR and RWCR are strongly correlated with each other (0.63 to 0.75), showing that they are related but different measures. For RWCR-10K, the strong correlation with WCR and WER suggests that the choice of 10,000 as a frequency cut-off has validity.

6 Improvements and further directions

6.1 Improving recognition of common words

The analysis of the recognition results presented in Chapter 5 shows that the topic-adapted language models are less successful than the reference model at recognizing more common words, especially the top 1000 by frequency rank (for example as shown in Figure 5-3). This clearly impacts on readability of the resulting transcript. As an ideal result would be to generate transcripts which are both more searchable and more readable than with the generic reference model, improving recognition rates for common words is a worthwhile goal.

Analysis of differential word recognition rates (for example as in Table 5-15) shows that recognition failures of words in this group are not a question of vocabulary (i.e. whether the words are included in the language model or not), but are a consequence of the language model's construction.

Areas to investigate include:

- Altering the size and shape of the language model, in particular the number of unigrams (vocabulary size), and the number of trigrams which can be limited by applying a frequency cut-off. A key question is whether excluding low-frequency trigrams would negatively impact recognition of specialist vocabulary.
- Adjusting the size and composition of the input corpus: for example using a larger set of sentences for the generic language model, adjusting the vocabulary balance between the generic and topic-derived language model, and the method of interpolation used to create the topic-adapted language model.
- Closer alignment of genre: HUB4 is derived from broadcast news transcripts, which in some respects may be closer in genre to spoken lectures than Wikipedia articles. A collection of lecture speech transcripts (ideally verbatim transcripts so that disfluences and repetitions are more accurately modelled) could be used as an additional source of language modelling data for the topic-adapted language models.

6.2 Iterative similarity modelling

The current approach generates a language model for a topic by proceeding from lecture metadata (such as title) to keywords, from keywords to seed articles, and from seed articles to related articles. Article similarity is further determined transitively: if A is similar to B and B is similar to C then A is similar to C. While this may be a weak assumption, the strength of the relationship between the seed articles and the lecture contents is also unknown, so the method uses a broad and coarse net to gather possibly related articles.

However, once a transcript has been created, it is possible to short-circuit the chain of inferences above and re-run the article harvesting process against a large set of articles, calculating similarity to the hypothesis transcript directly rather than to the parent article. This process could be repeated iteratively, as illustrated in Figure 6-1, each time generating a progressively more adapted language model.

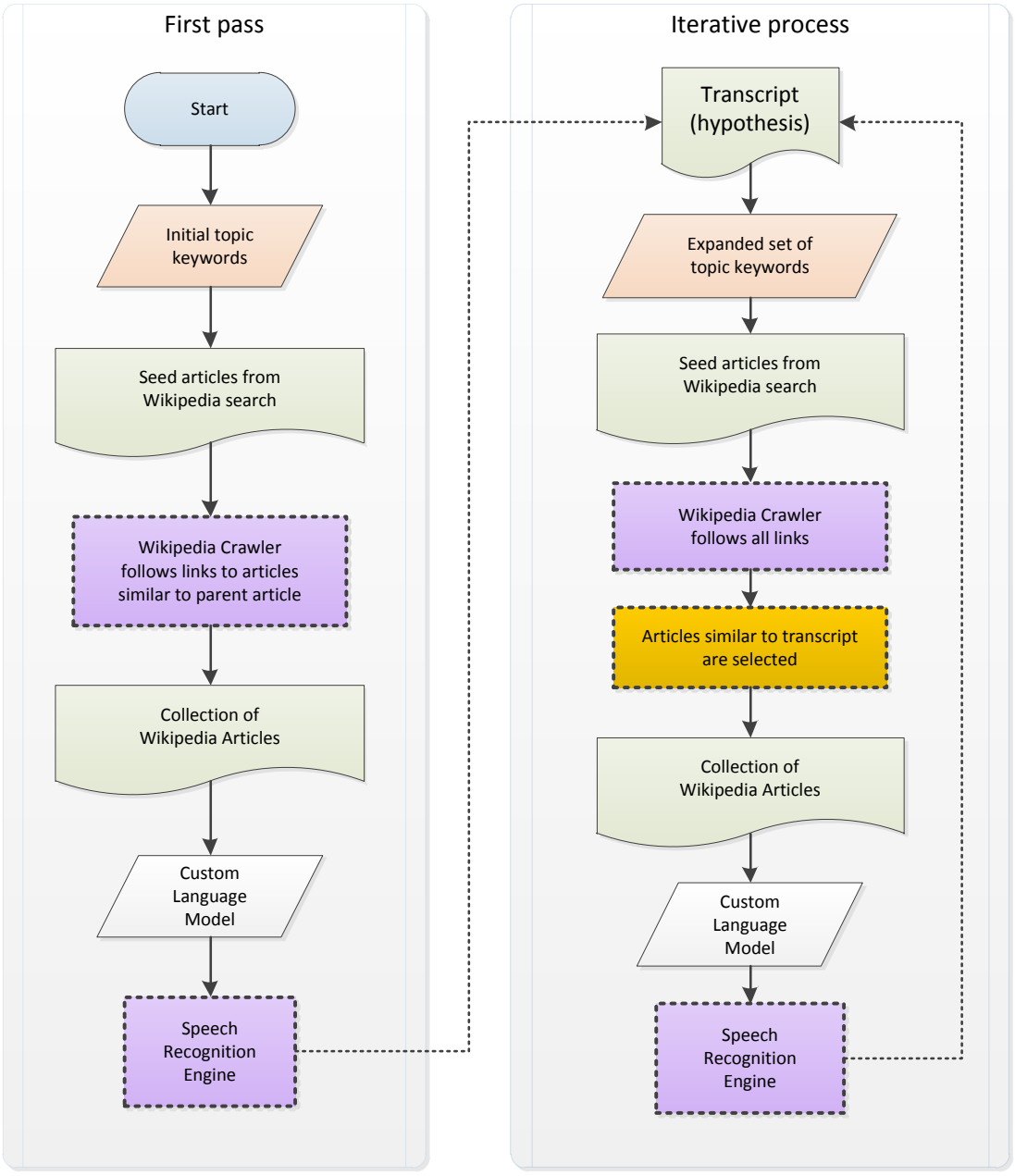


Figure 6-1: Iterative Similarity Modelling

6.3 Improving pronunciation accuracy

Table 5-12 shows that the recognition rates of words with estimated pronunciation is significantly poorer than for words contained in the CMU Dictionary. Approaches which could yield improvements include:

- Increasing the size of the pronunciation dictionary, i.e. adding manually vetted pronunciations to the dictionary.
- Including more than one pronunciation hypothesis in the dictionary.
- Identifying words and names which may originate from other languages, and using the appropriate (non-English) pronunciation dictionary, or estimated pronunciation using a model trained from the appropriate language.

6.4 Generalizability to other languages

Although the approach described here has been investigated only for lectures in English, the technique should in theory be generalizable to other languages. The main constraint is likely to be the size of the Wikipedia for the target language. While English Wikipedia is currently the largest and projected to plateau at around 4.4 million articles, 9 other Wikipedias have in excess of 750,000 articles, and seem likely to continue to grow over time, as illustrated in Figure 6-2 [78].

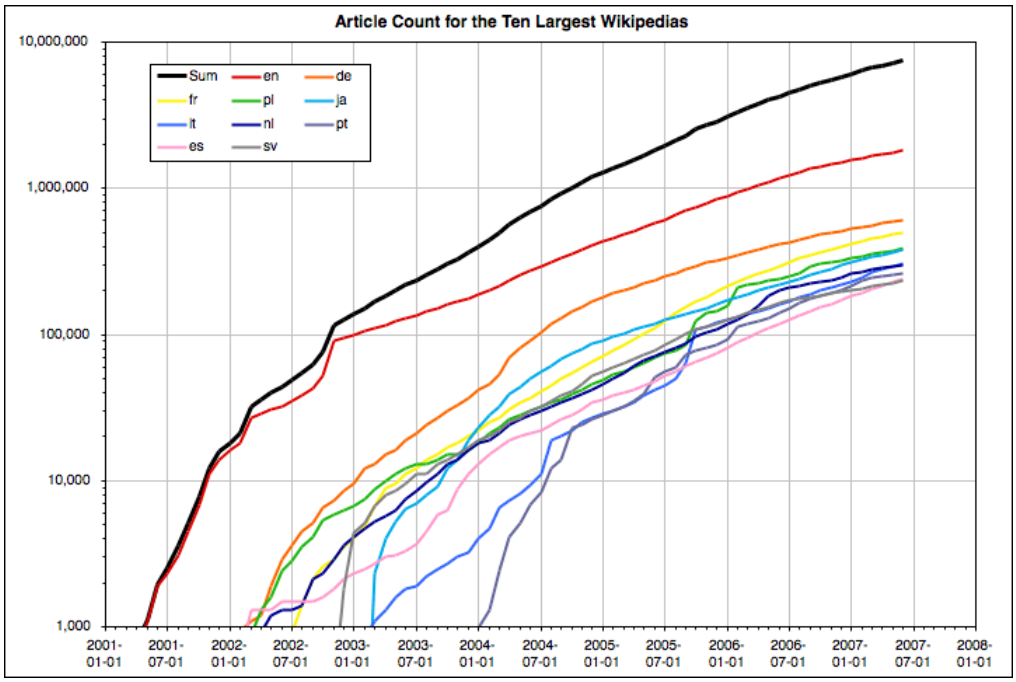


Figure 6-2: Article Count for the Ten Largest Wikipedias, 2001-2007

Other language-specific resources and requirements include:

- A pronouncing dictionary and model for g2p estimation
- Sentence boundary detection rules or model
- Punctuation rules or other constraints for text conditioning
- UTF8 support in all tools.

6.5 Examining user search behaviour

The methodology and evaluation followed here have rested on the assumption that users are likely to search with less-common terms, and that these can be characterised in general by their position in a frequency-ranked dictionary.

Applying the discipline of user-centred design to examine user search behaviour may provide insights as to how users construct search terms for particular goals and navigate the results, and thus possibly generate new approaches to optimizing recognition performance to improve search success.

7 Conclusions

7.1 Implementation and analysis

To investigate the research questions, a prototype system was developed to generate topic-adapted dictionaries and language models from Wikipedia for each of a set sample lectures from Open Yale Courses.

Two strategies for extracting topic-specific subsets of Wikipedia articles were investigated: a naïve crawler following all article links from a set of seed articles, and a refinement following only links to articles sufficiently similar to the parent article. Pair-wise article similarity was computed using latent semantic indexing.

The transcripts produced by the CMU Sphinx4 ASR engine using the Wikipedia topic-adapted models were compared to those produced by the reference HUB4 language model, evaluated across a number of standard metrics and one new metric.

7.2 Increasing transcript searchability with topic-adapted language models created from Wikipedia articles harvested by the Similarity Crawler

Returning to the research questions posed in 1.8, the first sub-question is:

> *To what extent do topic-adapted language models created from Wikipedia produce more searchable transcripts than those created using a generic reference language model?*

With the methodology described in Chapters 3 and 4, the evaluation results discussed in Chapter 5 show that topic-adapted language models created from Wikipedia articles harvested by the Similarity Crawler produce results of marginally worse accuracy than those produced by the reference HUB4 language model (Table 5-10), with an average increase in Word Error Rate (WER) of 0.8% absolute, and an average decrease in Word Correct Rate (WCR) of 3.3% absolute.

However, examination of the accuracy of the topic-adapted language models by word frequency rank shows that the topic-adapted language models increase recognition accuracy for less frequent words, while decreasing accuracy for more frequent words (Figure 5-3).

The notion of "searchability" is formalized through the metric Ranked Word Correct Rate (RWCR), following the assumption that searchability is associated with recall (word accuracy) for less frequent words. Using the metric RWCR-10K (Ranked Word Correct Rate for words below 10,000 in a frequency-ranked dictionary), the topic-adapted language models created using the Wikipedia Similarity Crawler outperform the reference HUB4 language model by an average of 9% absolute (Table 5-18).

Thus while the topic-adapted language models produce transcripts which are less accurate overall (and thus less readable), the improvement in RWCR suggests that they are more searchable than those produced by the HUB4 reference language model.

A positive relationship between RWCR and searchability is assumed here. However, this hypothesis should be confirmed through further research such as user trials, which fell outside the scope of this preliminary study.

7.3 Assessing the effectiveness of an article similarity metric when creating topic-adapted language models using a Wikipedia article crawler

The second sub-question identified in 1.8 is:

> *To what extent do topic-adapted language models created from Wikipedia using a crawler strategy bound by an article similarity metric produce more searchable transcripts than those created from Wikipedia using a naïve crawler strategy?*

Using article similarity to constrain the links followed by a Wikipedia crawler produces an average net improvement in the number of out-of-vocabulary (OOV) words, Perplexity, WER, WCR and RWCR-10K scores for the transcripts produced using the resulting language model.

However, this effect is not even across all the lectures evaluated. For OOV, Perplexity, WER and WCR, language models using the Similarity Crawler produced better results across all metrics than language models using the Naïve Crawler for only 7 of the 13 lectures (Table 5-8). For RWCR-10K, Similarity Crawler language models produced better results than Naïve Crawler language models for 8 of the 13 lectures (Table 5-8).

Most of the benefit provided by the Wikipedia-derived language models therefore comes from the initial keyword-based search (8% average improvement in RWCR-10K), with the article similarity score only providing a relatively small further gain (1% additional average improvement in RWCR-10K).

A technique for deriving increased benefit from article similarity using an iterative language modelling and recognition process is suggested in 6.2.

7.4 Overall

The overall research question is:

> *How can English Wikipedia be used as a language corpus for the unsupervised topic adaptation of language models to improve the searchability of lecture transcripts generated by an automatic speech recognition engine?*

Chapter 4 describes a process for using Wikipedia to generate topic-adapted language models for a speech recognition engine to improve the searchability of lecture transcripts.

The evaluation results in Chapter 5 show the resulting transcripts are less accurate than those produced by the reference language model, but are potentially more searchable as they have greater accuracy with respect to less frequent words which are more likely to be used as search terms.

Chapter 6 suggests avenues for investigation to improve accuracy further, so that it is not necessary to trade off readability for searchability. Possible strategies include adjusting how the language models are created to improve their quality, using iterative similarity modelling to further refine the language model from the initial hypothesis, and improving pronunciation accuracy.

References

[1] Jan-Martin Lowendahl, "Hype Cycle for Education, 2011." Gartner, Inc, 29-Jul-2011.

[2] J. Copley, "Audio and video podcasts of lectures for campus-based students: production and evaluation of student use," *Innovations in Education and Teaching International*, vol. 44, no. 4, pp. 387–399, 2007.

[3] M. Ketterl, O. A. Schulte, and A. Hochman, "Opencast Matterhorn: A Community-Driven Open Source Solution for Creation, Management and Distribution of Audio and Video in Academia," in *Multimedia, International Symposium on*, Los Alamitos, CA, USA, 2009, pp. 687–692.

[4] Opencast Matterhorn Project, "Matterhorn Features." [Online]. Available: http://opencast.org/matterhorn/features. [Accessed: 19-Feb-2012].

[5] J. R. Glass, T. J. Hazen, D. S. Cyphers, K. Schutte, and A. Park, "The MIT spoken lecture processing project," in *Proceedings of HLT/EMNLP on Interactive Demonstrations*, 2005, pp. 28–29.

[6] MIT Computer Science and Artificial Intelligence Laboratory, "MIT Lecture Browser," *MIT Lecture Browser*. [Online]. Available: http://web.sls.csail.mit.edu/lectures/. [Accessed: 19-Feb-2012].

[7] C. Munteanu, G. Penn, R. Baecker, E. Toms, and D. James, "Measuring the acceptable word error rate of machine-generated webcast transcripts," in *Ninth International Conference on Spoken Language Processing*, 2006.

[8] Peter Berman, *Chemical Pathology of the Liver*. University of Cape Town: Lecture to Year 2 MBChB Students, 2010.

[9] Lewis Carroll, *Alice's Adventures in Wonderland*. Project Gutenberg, 2008.

[10] Marquard, Stephen, "Recognizing specialized vocabulary with large dictionaries," *Truly Madly Wordly: Open source language modelling and speech recognition*, 25-Mar-2011. [Online]. Available: http://trulymadlywordly.blogspot.com/2011/03/recognizing-specialist-vocabulary.html.

[11] J. Gauvain, G. Adda, L. Lamel, and M. Adda-Decker, "Transcribing broadcast news: The limsi nov96 hub4 system," in *Proc. ARPA Speech Recognition Workshop*, 1997, vol. 56, p. 63.

[12] L. Rabiner and B. H. Juang, *Fundamentals of speech recognition*. Prentice-Hall Englewood Cliffs, New Jersey, 1993.

[13] J. Baker, L. Deng, J. Glass, S. Khudanpur, C. H. Lee, N. Morgan, and D. O'Shaugnessy, "Research developments and directions in speech recognition and understanding. Part 1," *IEEE Signal Processing Magazine*, vol. 26, no. 3, pp. 75–80, 2009.

[14] I. P. Association, *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge Univ Pr, 1999.

[15] V. W. Zue and S. Seneff, "Transcription and alignment of the TIMIT database," in *Proceedings of the second meeting on advanced man-machine interface through spoken language*, 1988, pp. 11–1.

[16] Carnegie Mellon University, "The CMU Pronouncing Dictionary." [Online]. Available: http://www.speech.cs.cmu.edu/cgi-bin/cmudict/. [Accessed: 19-Jul-2012].

[17] D. B. Paul and J. M. Baker, “The design for the wall street journal-based CSR corpus,” in *Proceedings of the workshop on Speech and Natural Language - HLT ’91*, Harriman, New York, 1992, p. 357.

[18] Jonathan Fiscus, John Garofolo, Mark Przybocki, William Fisher, and David Pallett, “1997 English Broadcast News Speech (HUB4).” Linguistic Data Consortium, Philadelphia.

[19] E. Leeuwis, M. Federico, and M. Cettolo, “Language modeling and transcription of the TED corpus lectures,” 2003. [Online]. Available: http://doc.utwente.nl/55870/. [Accessed: 05-Nov-2010].

[20] C. Munteanu, “Useful Transcriptions of Webcast Lectures,” PhD Thesis, University of Toronto, 2009.

[21] S. Furui, K. Maekawa, H. Isahara, T. Shinozaki, and T. Ohdaira, “Toward the realization of spontaneous speech recognition-introduction of a Japanese priority program and preliminary results,” in *Sixth International Conference on Spoken Language Processing*, 2000.

[22] K. Bain, S. H. Basson, and M. Wald, “Speech recognition in university classrooms: Liberated Learning Project,” in *Proceedings of the fifth international ACM conference on Assistive technologies - Assets ’02*, Edinburgh, Scotland, 2002, p. 192.

[23] E. Luppi, R. Primiani, C. Raffaelli, D. Tibaldi, I. Traina, and A. Violi, “Net4voice: new technologies for voice-converting in barrier-free learning environments (Final Report),” Net4Voice Project, 135446-LLP-1-2007-1-IT-KA3-KA3MP, Jul. 2010.

[24] L. Lamel, G. Adda, E. Bilinski, and J. L. Gauvain, “Transcribing lectures and seminars,” in *Ninth European Conference on Speech Communication and Technology*, 2005.

[25] H. Yamazaki, K. Iwano, K. Shinoda, S. Furui, and H. Yokota, “Dynamic language model adaptation using presentation slides for lecture speech recognition,” in *Proc. Interspeech*, 2007, pp. 2349–2352.

[26] J. Glass, T. J. Hazen, L. Hetherington, and C. Wang, “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*, 2004, pp. 9–12.

[27] Y. Akita and T. Kawahara, “Statistical Transformation of Language and Pronunciation Models for Spontaneous Speech Recognition,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 6, pp. 1539–1549, 2010.

[28] J. P. Birnholtz, “Back to school: Design principles for improving webcast interactivity from face-to-face classroom observation,” in *Proc. DIS*, 2006, vol. 6, pp. 311–320.

[29] X. Anguera Miro, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, “Speaker diarization: A review of recent research,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 2, pp. 356–370, 2012.

[30] M. Cettolo, F. Brugnara, and M. Federico, “Advances in the automatic transcription of lectures,” in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP ’04). IEEE International Conference on*, 2004, vol. 1, p. I–769–72 vol.1.

[31] H. Nanjo and T. Kawahara, “Language model and speaking rate adaptation for spontaneous presentation speech recognition,” *Speech and Audio Processing, IEEE Transactions on*, vol. 12, no. 4, pp. 391–400, 2004.

[32] T. J. Hazen and E. McDermott, “Discriminative MCE-Based Speaker Adaptation of Acoustic Models for a Spoken Lecture Processing Task,” presented at the Interspeech 2007, Antwerp, Belgium, 2007, pp. 1577 – 1580.

[33] T. Niesler and D. Willett, “Language identification and multilingual speech recognition using discriminatively trained acoustic models,” in *Multilingual Speech and Language Processing*, 2006.

[34] K. Kato, H. Nanjo, and T. Kawahara, “Automatic transcription of lecture speech using topic-independent language modeling,” in *Sixth International Conference on Spoken Language Processing*, 2000.

[35] T. Niesler and D. Willett, “Unsupervised language model adaptation for lecture speech transcription,” presented at the 7th International Conference on Spoken Language Processing, Denver, Colorado, 2002, vol. 144, p. 413K.

[36] D. Willettt, T. Niesler, E. McDermott, Y. Minami, and S. Katagiri, “Pervasive unsupervised adaptation for lecture speech transcription,” in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP ’03). 2003 IEEE International Conference on*, 2003, vol. 1, p. I–292–I–295 vol.1.

[37] S. Della Pietra, V. Della Pietra, R. L. Mercer, and S. Roukos, “Adaptive language modeling using minimum discriminant estimation,” in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, 1992, vol. 1, pp. 633–636.

[38] H. Nanjo and T. Kawahara, “Unsupervised language model adaptation for lecture speech recognition,” in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[39] C. Munteanu, G. Penn, and R. Baecker, “Web-based language modelling for automatic lecture transcription,” in *Proceedings of 8th Annual Conference of the International Speech Communication Association*, 2007, pp. 2353–2356.

[40] T. Kawahara, Y. Nemoto, and Y. Akita, “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*, 2008, pp. 4929–4932.

[41] F. Wood, J. Gasthaus, C. Archambeau, L. James, and Y. W. Teh, “The sequence memoizer,” *Communications of the ACM*, vol. 54, no. 2, pp. 91–98, 2011.

[42] H. Bourlard, H. Hermansky, and N. Morgan, “Towards increasing speech recognition error rates,” *Speech Communication*, vol. 18, no. 3, pp. 205–231, May 1996.

[43] I. Mccowan, D. Moore, J. Dines, D. Gatica-Perez, M. Flynn, and P. Wellner, “On the Use of Information Retrieval Measures for Speech Recognition Evaluation,” *IDIAP*, no. Idiap-RR-73–2004, 2005.

[44] A. Park, T. J. Hazen, and J. R. Glass, “Automatic Processing of Audio Lectures for Information Retrieval: Vocabulary Selection and Language Modeling,” in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP ’05). IEEE International Conference on*, 2005, vol. 1, pp. 497–500.

[45] Ye-Yi Wang, A. Acero, and C. Chelba, “Is word error rate a good indicator for spoken language understanding accuracy,” in *2003 IEEE Workshop on Automatic Speech Recognition and Understanding, 2003. ASRU ’03*, 2003, pp. 577– 582.

[46] K. Rankin, R. Baecker, and P. Wolf, “ePresence; An Open Source Interactive Webcasting and Archiving System for eLearning,” in *Proceedings E-Learn*, 2004.

[47] Boulder Language Technologies, “Sonic: Large Vocabulary Continuous Speech Recognition System,” 2012. [Online]. Available: http://www.bltek.com/virtual-teacher-side-menu/sonic.html. [Accessed: 26-Feb-2012].

[48] B. Pellom and K. Hacioglu, “Recent improvements in the CU SONIC ASR system for noisy speech: The SPINE task,” in *Acoustics, Speech, and Signal Processing, 2003.*

Proceedings.(ICASSP'03). 2003 IEEE International Conference on, 2003, vol. 1, p. I–4.

[49] C. Munteanu, R. Baecker, and G. Penn, "Collaborative editing for improved usefulness and usability of transcript-enhanced webcasts," in *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, Florence, Italy, 2008, p. 373.

[50] C. Munteanu, G. Penn, and X. Zhu, "Improving automatic speech recognition for lectures through transformation-based rules learned from minimal data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, 2009, pp. 764–772.

[51] C. Munteanu, R. Baecker, G. Penn, E. Toms, and D. James, "The effect of speech recognition accuracy rates on the usefulness and usability of webcast archives," in *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*, Montreal, Qubec, Canada, 2006, p. 493.

[52] MIT Computer Science and Artificial Intelligence Laboratory, "SUMMIT Speech Recognizer," *SLS :: Research Initiatives :: Technologies :: SUMMIT*. [Online]. Available: http://groups.csail.mit.edu/sls/technologies/asr.shtml.

[53] J. Glass, J. Chang, and M. McCandless, "A probabilistic framework for feature-based speech recognition," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, 1996, vol. 4, pp. 2277–2280.

[54] B. Muramatsu, "Automated Lecture Transcription," presented at the OCW Consortium Global Meeting, 2009.

[55] J. Glass, T. J. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in the MIT spoken lecture processing project," in *Proc. Interspeech*, 2007, vol. 3.

[56] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 1, pp. 186–197, 2007.

[57] G. T. Yu, "Efficient Error Correction for Speech Recognition Systems using Constrained Re-recognition," Thesis, Massachusetts Institute of Technology, 2008.

[58] I. Badr, I. McGraw, and J. Glass, "Pronunciation Learning from Continuous Speech," presented at the INTERSPEECH 2011, Florence, Italy, 2011, pp. 549 – 552.

[59] S. Liu, S. Seneff, and J. Glass, "A collective data generation method for speech language models," in *Spoken Language Technology Workshop (SLT), 2010 IEEE*, 2010, pp. 223–228.

[60] C. Lee and J. Glass, "A Transcription Task for Crowdsourcing with Automatic Quality Control," *Proc. Interspeech2011, Florence*, 2011.

[61] M. Wald, "Using Automatic Speech Recognition to Enhance Education for All Students: Turning a Vision into Reality," in *Frontiers in Education, 2005. FIE '05. Proceedings 35th Annual Conference*, 2005, p. S3G.

[62] O. A. Schulte, T. Wunden, and A. Brunner, "REPLAY: an integrated and open solution to produce, handle, and distribute audio-visual (lecture) recordings," in *Proceedings of the 36th annual ACM SIGUCCS fall conference: moving mountains, blazing trails*, 2008, pp. 195–198.

[63] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel, "Sphinx-4: a flexible open source framework for speech recognition," Sun Microsystems, Inc., Mountain View, CA, USA, 2004.

[64] S. Atitallah, “Speech indexation in Replay,” Bachelor’s thesis, ETH Zurich, 2009.

[65] R. Kheir and T. Way, “Inclusion of deaf students in computer science classes using real-time speech transcription,” in *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education - ITiCSE ’07*, Dundee, Scotland, 2007, p. 261.

[66] M. Wald, “Synote: Accessible and Assistive Technology Enhancing Learning for All Students,” *Computers Helping People with Special Needs*, pp. 177–184, 2010.

[67] M. Lin, J. F. Nunamaker Jr, M. Chau, and H. Chen, “Segmentation of lecture videos based on text: a method combining multiple linguistic features,” in *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, 2004, p. 9.

[68] T. Kawahara, M. Hasegawa, K. Shitaoka, T. Kitade, and H. Nanjo, “Automatic indexing of lecture presentations using unsupervised learning of presumed discourse markers,” *Speech and Audio Processing, IEEE Transactions on*, vol. 12, no. 4, pp. 409–419, 2004.

[69] F. Seide, P. Yu, C. Ma, and E. Chang, “Vocabulary-independent search in spontaneous speech,” in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP’04). IEEE International Conference on*, 2004, vol. 1.

[70] C. W. Ngo, F. Wang, and T. C. Pong, “Structuring lecture videos for distance learning applications,” in *Multimedia Software Engineering, 2003. Proceedings. Fifth International Symposium on*, 2005, pp. 215–222.

[71] S. Repp and C. Meinel, “Semantic indexing for recorded educational lecture videos,” in *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW’06)*, 2006.

[72] S. Repp, A. Gross, and C. Meinel, “Browsing within Lecture Videos Based on the Chain Index of Speech Transcription,” *Learning Technologies, IEEE Transactions on*, vol. 1, no. 3, pp. 145–156, 2008.

[73] F. Liu, D. Pennell, F. Liu, and Y. Liu, “Unsupervised approaches for automatic keyword extraction using meeting transcripts,” in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics on ZZZ*, 2009, pp. 620–628.

[74] T. J. Hazen, “Automatic alignment and error correction of human generated transcripts for long speech recordings,” in *Ninth International Conference on Spoken Language Processing*, 2006.

[75] D. Tibaldi, “Speech recognition supporting learning: The future,” Net4Voice Project, 2010.

[76] CMU Sphinx Project, “HUB4 Open Source US English Acoustic and Language Models.” [Online]. Available: http://www.speech.cs.cmu.edu/sphinx/models/hub4opensrc_jan2002/INFO_ABOUT_MODELS.

[77] Wikipedia contributors, “Wikipedia:Size comparisons,” *Wikipedia, the free encyclopedia*. Wikimedia Foundation, Inc., 23-Jul-2012.

[78] Wikipedia contributors, “Wikipedia:Size of Wikipedia,” *Wikipedia, the free encyclopedia*. Wikimedia Foundation, Inc., 27-Jul-2012.

[79] Robert Parker, David Graff, Junbo Kong, Ke Chen and Kazuaki Maeda, “English Gigaword Fifth Edition.” Linguistic Data Consortium, Philadelphia, 2011.

[80] S. P. Ponzetto and M. Strube, “Knowledge derived from Wikipedia for computing semantic relatedness,” *Journal of Artificial Intelligence Research*, vol. 30, no. 1, pp. 181–212, 2007.

[81] Wikipedia contributors, “Wikipedia:Copyrights,” *Wikipedia, the free encyclopedia*. Wikimedia Foundation, Inc., 13-Jul-2012.

[82] Stephen Marquard, “Creating a text corpus from Wikipedia,” *Truly Madly Wordly: Open source language modelling and speech recognition*, 15-Mar-2011. [Online]. Available: http://trulymadlywordly.blogspot.com/2011/03/creating-text-corpus-from-wikipedia.html.

[83] J. Novak, D. Yang, N. Minematsu, and K. Hirose, “Initial and Evaluations of an Open Source WFST-based Phoneticizer,” The University of Tokyo, Tokyo Institute of Technology.

[84] L. Galescu and J. F. Allen, “Bi-directional conversion between graphemes and phonemes using a joint n-gram model,” in *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*, 2001.

[85] T. K. Landauer, P. W. Foltz, and D. Laham, “An introduction to latent semantic analysis,” *Discourse Processes*, vol. 25, no. 2–3, pp. 259–284, 1998.

[86] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[87] R. Řehůřek and P. Sojka, “Software Framework for Topic Modelling with Large Corpora,” in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, Valletta, Malta, 2010, pp. 45–50.

[88] R. Řehůřek, “Experiments on the English Wikipedia,” 02-Dec-2011. [Online]. Available: http://radimrehurek.com/gensim/wiki.html. [Accessed: 09-Jan-2012].

[89] R. B. Bradford, “An empirical study of required dimensionality for large-scale latent semantic indexing applications,” in *Proceeding of the 17th ACM conference on Information and knowledge management*, 2008, pp. 153–162.

[90] J. B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, and others, “Quantitative analysis of culture using millions of digitized books,” *science*, vol. 331, no. 6014, p. 176, 2011.

[91] Google, Inc., “Google Books Ngram Viewer Datasets,” 2010. [Online]. Available: http://books.google.com/ngrams/datasets.

[92] Stephen Marquard, “Using freetts and phonetisaurus for creating custom Sphinx dictionaries,” *Truly Madly Wordly: Open source language modelling and speech recognition*, 16-May-2011. [Online]. Available: http://trulymadlywordly.blogspot.com/2011/05/using-freetts-and-phonetisaurus-for.html.

Appendix 1: Software and Data Sets

Open Source Software Toolkits

Sphinx-4 1.0 beta
https://cmusphinx.svn.sourceforge.net/svnroot/cmusphinx/trunk/sphinx4
Last Changed Rev: 10998
Last Changed Date: 2011-05-31 12:38:52 +0200 (Tue, 31 May 2011)

SphinxBase 0.7 (sphinx_lm_convert)
https://cmusphinx.svn.sourceforge.net/svnroot/cmusphinx/trunk/sphinxbase
Last Changed Rev: 11011
Last Changed Date: 2011-06-06 15:26:59 +0200 (Mon, 06 Jun 2011)

MIT Language Modeling Toolkit v0.4
http://mitlm.googlecode.com/svn/trunk
Last Changed Rev: 48
Last Changed Date: 2010-11-29 23:59:06 +0200 (Mon, 29 Nov 2010)

Phonetisaurus grapheme-to-phoneme (g2p) framework
http://code.google.com/p/phonetisaurus/
Revision 895e2ba6b4b0, 28 May 2011

Gensim topic modelling toolkit, v0.7.8
http://radimrehurek.com/gensim/
https://github.com/piskvorky/gensim/zipball/0.7.8

gwtwiki toolkit, v3.0.16
http://code.google.com/p/gwtwiki/
Used for parsing Wikimedia markup to plain text.

OpenNLP toolkit, v1.5.1-incubating
http://incubator.apache.org/opennlp/
Used for sentence boundary detection.

Sphinx HUB4 Models

HUB4 acoustic model
http://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/US%20English%20HUB4%20Acoustic%20Model/

HUB-4 Binary Trigram Language Model, in HUB4_trigram_lm.zip available from
http://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/US%20English%20HUB4%20Language%20Model/

Language Resources

Wikipedia full text dump, enwiki-latest-pages-articles.xml.bz2 as at 15 Feb 2011
downloaded from http://download.wikimedia.org/enwiki/latest/

Google books n-gram dataset (English 20090715)
http://books.google.com/ngrams/datasets

The CMU Pronouncing Dictionary, version 0.7a, available from
https://cmusphinx.svn.sourceforge.net/svnroot/cmusphinx/trunk/cmudict/

FST phonetisaurus model trained from CMUdict 0.7a, available from
http://www.gavo.t.u-tokyo.ac.jp/~novakj/cmudict.0.7a.tgz

opennlp English sentence detector model
http://opennlp.sourceforge.net/models-1.5/en-sent.bin

Appendix 2: Source code

Selected source code used in this project is available from the svn repository maintained by the Centre for Educational Technology, UCT. The links below are to the versions of the code used in the project with the specific versions of the packages listed in Appendix 1.

Source code and scripts are licensed under the Apache 2.0 license (http://www.apache.org/licenses/LICENSE-2.0.html) except where noted otherwise.

Java and python code for Wikipedia plain text export and Wikipedia Crawler:
http://source.cet.uct.ac.za/svn/people/smarquard/wikicrawler/trunk/?p=10355

Scripts used to generate the custom language models, set up recognition jobs, evaluate the output and calculate metrics:
http://source.cet.uct.ac.za/svn/people/smarquard/sphinx/scripts/?p=10355

Appendix 3: Open Yale Courses lectures

Selected Lectures

Audio recordings and transcripts of lectures from Open Yale Courses at http://oyc.yale.edu/ used in accordance with Terms of Use described at http://oyc.yale.edu/terms:

Charles Bailyn, Frontiers and Controversies in Astrophysics (Yale University: Open Yale Courses), http://oyc.yale.edu (Accessed June 8, 2011). License: Creative Commons BY-NC-SA

Mark Saltzman, Frontiers of Biomedical Engineering (Yale University: Open Yale Courses), http://oyc.yale.edu (Accessed June 8, 2011). License: Creative Commons BY-NC-SA

Stephen Stearns, Principles of Evolution, Ecology and Behavior (Yale University: Open Yale Courses), http://oyc.yale.edu (Accessed June 8, 2011). License: Creative Commons BY-NC-SA

John Rogers, Milton (Yale University: Open Yale Courses), http://oyc.yale.edu (Accessed June 8, 2011). License: Creative Commons BY-NC-SA

Amy Hungerford, The American Novel Since 1945 (Yale University: Open Yale Courses), http://oyc.yale.edu (Accessed June 8, 2011). License: Creative Commons BY-NC-SA

Paul Fry, Introduction to Theory of Literature (Yale University: Open Yale Courses), http://oyc.yale.edu (Accessed June 8, 2011). License: Creative Commons BY-NC-SA

Joanne Freeman, The American Revolution (Yale University: Open Yale Courses), http://oyc.yale.edu (Accessed June 8, 2011). License: Creative Commons BY-NC-SA

John Merriman, European Civilization, 1648-1945 (Yale University: Open Yale Courses), http://oyc.yale.edu (Accessed June 8, 2011). License: Creative Commons BY-NC-SA

Shelly Kagan, Death (Yale University: Open Yale Courses), http://oyc.yale.edu (Accessed June 8, 2011). License: Creative Commons BY-NC-SA

Steven Smith, Introduction to Political Philosophy (Yale University: Open Yale Courses), http://oyc.yale.edu (Accessed June 8, 2011). License: Creative Commons BY-NC-SA

Paul Bloom, Introduction to Psychology (Yale University: Open Yale Courses), http://oyc.yale.edu (Accessed June 8, 2011). License: Creative Commons BY-NC-SA

Dale Martin, Introduction to New Testament History and Literature (Yale University: Open Yale Courses), http://oyc.yale.edu (Accessed June 8, 2011). License: Creative Commons BY-NC-SA

Lecture Transcripts

Reference transcripts from the selected Open Yale Courses lectures (derivative works in terms of the Creative Commons BY-NC-SA license) are archived at:
http://source.cet.uct.ac.za/svn/people/smarquard/datasets/oyc-transcripts/

Source URLs for the lecture transcripts are:

http://oyc.yale.edu/astronomy/frontiers-and-controversies-in-astrophysics/content/sessions/lecture21.html

http://oyc.yale.edu/biomedical-engineering/frontiers-in-biomedical-engineering/content/sessions/session-5-cell-culture-engineering

http://oyc.yale.edu/biomedical-engineering/frontiers-in-biomedical-engineering/content/sessions/session-9-biomolecular-engineering-engineering-of

http://oyc.yale.edu/ecology-and-evolutionary-biology/principles-of-evolution-ecology-and-behavior/content/sessions/lecture34.html

http://oyc.yale.edu/english/milton/content/sessions/session-6-lycidas

http://oyc.yale.edu/english/american-novel-since-1945/content/sessions/session-12-thomas-pynchon-the-crying-of-lot-49

http://oyc.yale.edu/english/introduction-to-theory-of-literature/content/sessions/lecture15.html

http://oyc.yale.edu/history/the-american-revolution/content/sessions/lecture08.html

http://oyc.yale.edu/history/european-civilization-1648-1945/content/sessions/lecture06.htm

http://oyc.yale.edu/philosophy/death/content/sessions/lecture13.html

http://oyc.yale.edu/political-science/introduction-to-political-philosophy/content/sessions/lecture02.html

http://oyc.yale.edu/yale/psychology/introduction-to-psychology/content/sessions/lecture05.html

http://oyc.yale.edu/religious-studies/introduction-to-new-testament/content/sessions/lecture26.html

Appendix 4: Sphinx Configuration

http://source.cet.uct.ac.za/svn/people/smarquard/datasets/sphinx4-hub4-oyc/sphinx-custom.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Sphinx-4 Configuration file: HUB4 language model -->

<config>

    <!-- ******************************************************** -->
    <!-- frequently tuned properties                              -->
    <!-- ******************************************************** -->

    <property name="logLevel" value="INFO"/>

    <!-- Used in standardActiveListFactory: initial 30000 1E-60 -->
    <property name="absoluteBeamWidth" value="30000"/>
    <property name="relativeBeamWidth" value="1E-80"/>

    <!-- Used in wordActiveListFactory initial 22 1E-30 -->
    <property name="absoluteWordBeamWidth" value="22"/>
    <property name="relativeWordBeamWidth" value="1E-30"/>

    <!-- Used in LexTreeLinguist and LargeNGramModel -->
    <property name="languageWeight" value="10.5"/>
    <property name="wordInsertionProbability" value=".2"/>
    <property name="silenceInsertionProbability" value=".1"/>

    <!-- Component names -->
    <property name="frontend" value="epFrontEnd"/>
    <property name="recognizer" value="recognizer"/>

    <!-- ******************************************************** -->
    <!-- word recognizer configuration                            -->
    <!-- ******************************************************** -->

    <component name="recognizer" type="edu.cmu.sphinx.recognizer.Recognizer">
        <property name="decoder" value="decoder"/>
        <propertylist name="monitors">
        </propertylist>
    </component>

    <!-- ******************************************************** -->
    <!-- The Decoder   configuration                              -->
    <!-- ******************************************************** -->

    <component name="decoder" type="edu.cmu.sphinx.decoder.Decoder">
        <property name="searchManager" value="wordPruningSearchManager"/>
    </component>

    <!-- ******************************************************** -->
    <!-- The Search Manager                                       -->
    <!-- ******************************************************** -->

    <component name="wordPruningSearchManager"
type="edu.cmu.sphinx.decoder.search.WordPruningBreadthFirstSearchManager">
        <property name="scorer" value="threadedScorer"/>
        <property name="pruner" value="trivialPruner"/>
        <property name="acousticLookaheadFrames" value="1.8"/>
        <property name="logMath" value="logMath"/>
        <property name="activeListManager" value="activeListManager"/>
        <property name="buildWordLattice" value="false"/>
        <property name="relativeBeamWidth" value="1E-65"/>
        <property name="growSkipInterval" value="8"/>
        <property name="linguist" value="lexTreeLinguist"/>
        <property name="checkStateOrder" value="false"/>
```

```
        <property name="keepAllTokens" value="true"/>
    </component>

    <!-- ******************************************************** -->
    <!-- The Active Lists                                         -->
    <!-- ******************************************************** -->

    <component name="activeListManager"
type="edu.cmu.sphinx.decoder.search.SimpleActiveListManager">
        <propertylist name="activeListFactories">
            <item>standardActiveListFactory</item>
            <item>wordActiveListFactory</item>
            <item>wordActiveListFactory</item>
            <item>standardActiveListFactory</item>
            <item>standardActiveListFactory</item>
            <item>standardActiveListFactory</item>
        </propertylist>
    </component>

    <component name="standardActiveListFactory"
type="edu.cmu.sphinx.decoder.search.PartitionActiveListFactory">
        <property name="logMath" value="logMath"/>
        <property name="absoluteBeamWidth" value="${absoluteBeamWidth}"/>
        <property name="relativeBeamWidth" value="${relativeBeamWidth}"/>
    </component>

    <component name="wordActiveListFactory"
type="edu.cmu.sphinx.decoder.search.PartitionActiveListFactory">
        <property name="logMath" value="logMath"/>
        <property name="absoluteBeamWidth" value="${absoluteWordBeamWidth}"/>
        <property name="relativeBeamWidth" value="${relativeWordBeamWidth}"/>
    </component>

    <!-- ******************************************************** -->
    <!-- The Pruner                                               -->
    <!-- ******************************************************** -->

    <component name="trivialPruner"
type="edu.cmu.sphinx.decoder.pruner.SimplePruner"/>

    <!-- ******************************************************** -->
    <!-- TheScorer                                                -->
    <!-- ******************************************************** -->

    <component name="threadedScorer"
type="edu.cmu.sphinx.decoder.scorer.ThreadedAcousticScorer">
        <property name="frontend" value="${frontend}"/>
        <property name="isCpuRelative" value="true"/>
        <property name="numThreads" value="0"/>
        <property name="minScoreablesPerThread" value="10"/>
        <property name="scoreablesKeepFeature" value="true"/>
    </component>

    <!-- ******************************************************** -->
    <!-- The linguist  configuration                              -->
    <!-- ******************************************************** -->

    <component name="lexTreeLinguist"
type="edu.cmu.sphinx.linguist.lextree.LexTreeLinguist">
        <property name="wantUnigramSmear" value="true"/>
        <property name="wordInsertionProbability"
value="${wordInsertionProbability}"/>
        <property name="silenceInsertionProbability"
value="${silenceInsertionProbability}"/>
        <property name="fillerInsertionProbability" value=".2"/>
        <property name="unitInsertionProbability" value="1.0"/>
        <property name="addFillerWords" value="false"/>
        <property name="languageModel" value="ngramModel"/>
        <property name="languageWeight" value="${languageWeight}"/>
        <property name="logMath" value="logMath"/>
```

```
        <property name="dictionary" value="dictionary"/>
        <property name="unigramSmearWeight" value="1"/>
        <property name="cacheSize" value="0"/>
        <property name="generateUnitStates" value="false"/>
        <property name="acousticModel" value="hub4"/>
        <property name="unitManager" value="unitManager"/>
    </component>

    <!-- ******************************************************** -->
    <!-- The Dictionary configuration                             -->
    <!-- ******************************************************** -->

    <component name="dictionary"
type="edu.cmu.sphinx.linguist.dictionary.FastDictionary">
        <property name="dictionaryPath"
value="file:models/cmudict/cmudict.0.7a_SPHINX_40"/>
        <property name="fillerPath" value="file:models/wsj/noisedict"/>
        <property name="addenda" value="file:models/extradict/extra-hub4-saurus.dic"/>
        <property name="addSilEndingPronunciation" value="false"/>
        <property name="allowMissingWords" value="false"/>
        <property name="createMissingWords" value="true"/>
        <property name="wordReplacement" value="&lt;sil&gt;"/>
        <property name="unitManager" value="unitManager"/>
    </component>

    <!-- ******************************************************** -->
    <!-- The Language Model configuration                         -->
    <!-- ******************************************************** -->

    <component name="ngramModel"
type="edu.cmu.sphinx.linguist.language.ngram.large.LargeNGramModel">
        <property name="location" value="file:models/hub4-
lm/language_model.arpaformat.DMP"/>
        <property name="unigramWeight" value="0.7"/>
        <property name="maxDepth" value="3"/>
        <property name="logMath" value="logMath"/>
        <property name="dictionary" value="dictionary"/>
        <property name="wordInsertionProbability"
value="${wordInsertionProbability}"/>
        <property name="languageWeight" value="${languageWeight}"/>
    </component>

    <!-- ******************************************************** -->
    <!-- The acoustic model configuration                         -->
    <!-- ******************************************************** -->

    <component name="hub4"
type="edu.cmu.sphinx.linguist.acoustic.tiedstate.TiedStateAcousticModel">
        <property name="loader" value="sphinx3Loader"/>
        <property name="unitManager" value="unitManager"/>
    </component>

    <component name="sphinx3Loader"
type="edu.cmu.sphinx.linguist.acoustic.tiedstate.Sphinx3Loader">
        <property name="logMath" value="logMath"/>
        <property name="unitManager" value="unitManager"/>
        <property name="location" value="file:models/hub4opensrc.cd_continuous_8gau"/>
        <property name="dataLocation" value=""/>
    </component>

    <!-- ******************************************************** -->
    <!-- The unit manager configuration                           -->
    <!-- ******************************************************** -->

    <component name="unitManager"
type="edu.cmu.sphinx.linguist.acoustic.UnitManager"/>
```

```
    <!-- ******************************************************** -->
    <!-- The frontend configuration                               -->
    <!-- ******************************************************** -->

    <component name="epFrontEnd" type="edu.cmu.sphinx.frontend.FrontEnd">
        <propertylist name="pipeline">
            <item>audioFileDataSource</item>
            <item>dataBlocker </item>
            <item>speechClassifier </item>
            <item>speechMarker </item>
            <item>nonSpeechDataFilter </item>
            <item>premphasizer</item>
            <item>windower</item>
            <item>fft</item>
            <item>melFilterBank</item>
            <item>dct</item>
            <item>batchCMN</item>
            <item>featureExtraction</item>
        </propertylist>
    </component>

    <component name="audioFileDataSource"
type="edu.cmu.sphinx.frontend.util.AudioFileDataSource"/>
    <component name="dataBlocker" type="edu.cmu.sphinx.frontend.DataBlocker"/>
    <component name="speechClassifier"
type="edu.cmu.sphinx.frontend.endpoint.SpeechClassifier"/>
    <component name="speechMarker"
type="edu.cmu.sphinx.frontend.endpoint.SpeechMarker"/>
    <component name="nonSpeechDataFilter"
type="edu.cmu.sphinx.frontend.endpoint.NonSpeechDataFilter"/>
    <component name="premphasizer"
type="edu.cmu.sphinx.frontend.filter.Preemphasizer"/>
    <component name="windower"
type="edu.cmu.sphinx.frontend.window.RaisedCosineWindower"/>
    <component name="fft"
type="edu.cmu.sphinx.frontend.transform.DiscreteFourierTransform"/>

    <component name="melFilterBank"
type="edu.cmu.sphinx.frontend.frequencywarp.MelFrequencyFilterBank">
        <property name="minimumFrequency" value="133.3334"/>
        <property name="maximumFrequency" value="6855.4976"/>
        <property name="numberFilters" value="40"/>
    </component>

    <component name="dct"
type="edu.cmu.sphinx.frontend.transform.DiscreteCosineTransform"/>
    <component name="batchCMN" type="edu.cmu.sphinx.frontend.feature.BatchCMN"/>
    <component name="featureExtraction"
type="edu.cmu.sphinx.frontend.feature.DeltasFeatureExtractor"/>

    <!-- ******************************************************** -->
    <!--  Miscellaneous components                                -->
    <!-- ******************************************************** -->

    <component name="logMath" type="edu.cmu.sphinx.util.LogMath">
        <property name="logBase" value="1.0001"/>
        <property name="useAddTable" value="true"/>
    </component>
</config>
```

Glossary

AI	Artificial Intelligence
AM	Acoustic Model
API	Application Programming Interface
ARPA	Advanced Research Projects Agency
Arpabet	ARPA ASR Alphabet
ASCII	American Standard Code for Information Interchange (character set)
ASR	Automatic Speech Recognition
CHIL	Computers in the Human Interaction Loop, research project
CMU	Carnegie Mellon University
CMUdict	CMU Pronouncing Dictionary
CNN	Cable News Network (broadcaster)
CSAIL	Computer Science and Artificial Intelligence Laboratory at MIT
CSPAN	Cable-Satellite Public Affairs Network (broadcaster)
ETH Zürich	Eidgenössische Technische Hochschule Zürich, science and technology university
EU	European Union
FP6	Framework Project 6, a European Union funding programme
FST	Finite State Transducer
g2p	Grapheme-to-Phoneme
HMM	Hidden Markov Model
HTML	Hypertext Markup Language
HUB4	Language corpus from the 1996 DARPA/NIST Continuous Speech Recognition Broadcast News Hub-4 Benchmark Test, and derived acoustic and language models
IPA	International Phonetic Alphabet
IR	Information Retrieval
KHz	Kilohertz, measure of frequency and signal bandwidth
LDA	Latent Dirichlet Allocation
LL	Liberated Learning Project and Consortium
LM	Language Model
LSA	Latent Semantic Analysis
LSI	Latent Semantic Indexing

LVCSR	Large Vocabulary Continuous (or Connected) Speech Recognition
MDE	Minimum Discrimant Estimation
MIT	Massachusetts Institute of Technology
MP3	Moving Picture Experts Group (MPEG) Audio Layer 3, audio format
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NPR	National Public Radio (broadcaster)
OOV	Out-of-vocabulary words
OYC	Open Yale Courses
PCM	Pulse-code modulation, digital audio format
PLSA	Probabilistic Latent Semantic Analysis
REPLAY	Lecture capture system developed at ETH Zürich
RT	Runtime, execution time
RWCR	Ranked Word Correct Rate
SaaS	Software-as-a-service
SI	Speaker-independent
SONIC	ASR system developed at Colorado University
Sphinx4	ASR system developed at Carnegie Mellon University
SUMMIT	ASR system developed at CSAIL, MIT
TED	Translanguage English Database Corpus
TF-IDF	Term Frequency - Inverse Document Frequency
TIMIT	Transcribed American English corpus developed by Texas Instruments (TI) and Massachusetts Institute of Technology (MIT)
US / USA	United States of America
UTF8	Universal Character Set (UCS) Transformation Format – 8 bit, a Unicode character set
WAV	Waveform Audio File Format
WCR	Word Correct Rate
WER	Word Error Rate
WFST	Weighted Finite State Transducer
WSJ	Wall Street Journal, newspaper and derived language corpus
XML	Extensible Markup Language