Augmenting Automatic Speech Recognition and Search Models for Spoken Content Retrieval

Yasufumi Moriya

BA, MSc

A dissertation submitted in fulfilment of the requirements for the award of

Doctor of Philosophy (PhD)

to the

Dublin City University
School of Computing

Supervisor:
Prof. Gareth. J. F. Jones

June 2022

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D. is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Yoshifumi Moriya

(Candidate) ID No.: 12109223

Date: 25/08/2022

Contents

List of Figures

List of Tables

Augmenting Automatic Speech Recognition and Search Models for Spoken Content Retrieval

Yasufumi Moriya

Abstract

Spoken content retrieval (SCR) is a process to provide a user with spoken documents in which the user is potentially interested. Unlike textual documents, searching through speech is not trivial due to its representation. Generally, automatic speech recognition (ASR) is used to transcribe spoken content such as user-generated videos and podcast episodes into transcripts before search operations are performed. Despite recent improvements in ASR, transcription errors can still be present in automatic transcripts. This is in particular when ASR is applied to out-of-domain data or speech with background noise.

This thesis explores improvement of ASR systems and search models for enhanced SCR on user-generated spoken content. There are three topics explored in this thesis. Firstly, the use of multimodal signals for ASR is investigated. This is motivated to integrate background contexts of spoken content into ASR. Integration of visual signals and document metadata into ASR is hypothesised to produce transcripts more aligned to background contexts of speech. Secondly, the use of semi-supervised training and content genre information from metadata are exploited for ASR. This approach is motivated to mitigate the transcription errors caused by recognition of out-of-domain speech. Thirdly, the use of neural models and the model extension using N-best ASR transcripts are investigated. Using ASR N-best transcripts instead of 1-best for search models is motivated because "key terms" missed in 1-best can be present in the N-best transcripts.

A series of experiments are conducted to examine those approaches to improvement of ASR systems and search models. The findings suggest that semi-supervised training bring practical improvement of ASR systems for SCR and the use of neural ranking models in particular with N-best transcripts improve the result of known-item search over the baseline BM25 model.

Publications

Publications relating to the research investigation in Chapter 5.

- Y. Moriya and G. J. F. Jones. Lstm language model adaptation with images and titles for multimedia automatic speech recognition. In Proceedings of IEEE Workshop on Spoken Language Tecnology (SLT), pages 219-226, 2018.
- Y. Moriya and G. J. F. Jones. Multimodal speaker adaptation of acoustic model and language model for ASR using speaker face embedding. In Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 8643–8647, 2019.

Publications relating to the research investigation in Chapter 6

- Y. Moriya and G. J. F. Jones. An ASR Nbest Transcript Neural Ranking Model for Spoken Content Retrieval. In Proceedings of Automatic Speech Recognition and Understanding (ASRU), 2021.

Publications relating to the research investigation in Chapter 7

- Y. Moriya and G. J. F. Jones. An ASR Nbest Transcript Neural Ranking Model for Spoken Content Retrieval. In Proceedings of Automatic Speech Recognition and Understanding (ASRU), 2021.

Acknowledgments

First and foremost, I would like to express my gratitude to my supervisor, Gareth. J.F. Jones, for providing me with the opportunity to join this PhD journey. I am grateful for you to introduce me to the world of spoken search. I always admire your expert knowledge of Information Retrieval and hopefully I have learned a tiny fraction of your elegant and logical writing style. It was always a joyful experience to listen to the history of your research career, at MediaEval workshops, at the restaurant close to NII or anywhere else.

Working on a PhD project was a long journey and I could not make it without help of my old and new friends in Dublin and all over the world. I am thankful to Abhishek Kaushik for being a great colleague of mine and sharing similar PhD experiences with me, to Anastasia Papathanaki and Paul Hayes for sharing the house with me for 5 years including a dark pandemic time and for being such supportive friends with me, to Fiorenza Braconi for being supportive and understanding and delivering a bubble tea when I was busy writing this thesis, to Jamil Furqan Chaudry for playing games online with me and always asking me how my PhD is going, to Mateusz Dubiel for sharing academic life experiences and being friends for almost 8 years since our Master's programme, to Mateusz Westa for being in frequent contact with me despite being apart over the Atlantic, to my friends at ADAPT Centre for going to lunch together and exchanging ideas (good pre-pandemic time) and to my friends at Clontarf Lawn Tennis Club for teaching me the Irish culture and bringing me a bit of craic. I would also like to thank my family: my father, Takushi Moriya, and my sister, Kiho Moriya, for supporting my decision of pursuing a PhD. Hopefully, we will meet again soon as the restrictions start to ease in Japan.

Finally, I am grateful to the Science Foundation of Ireland for the generous research grants including an extension due to the worldwide pandemic. Without this support, I would have not been able to focus on my research project.

Chapter 1

Introduction

Recent years have seen a significant increase in the amount of spoken multimedia archives available on the Internet including user-generated videos and podcasts. The wide availability of high quality speech or video recording devices mean that amateur content creators can easily capture and share their spoken content on a sharing platform. As the size of the multimedia archives grows, however, users of spoken content sharing platforms can have difficulties finding content in which they are truly interested. Unlike textual documents for which the full content is easily searchable, spoken documents are generally searched by matching a user search query with titles and meta-data created by content creators. Titles and meta data descriptions of spoken content do not generally provide much information of the actual spoken content. The motivation for Spoken Content Retrieval (SCR) thus is to provide users with more sophisticated search functionalities able to return accurate fine granularity search of the spoken content itself (Larson and Jones, 2012).

The goal of an SCR system is, as with any other Information Retrieval (IR) systems, to obtain documents that satisfy user information needs. SCR systems generally operate on transcripts of the spoken content created using automatic speech recognition (ASR). Figure 1.1 illustrates the architecture of a modern SCR system. When a user query is submitted to the retrieval engine, the engine searches through

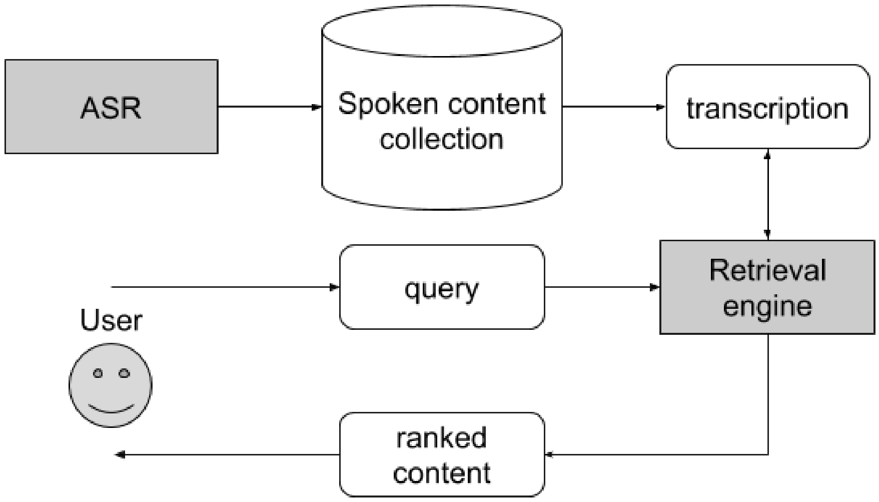


Figure 1.1: Basic architecture of an SCR system.

the collection of spoken content stored in the form of transcripts created by ASR. The task of the search engine is to assign a relevance score to each document in the collection and to return a ranked list of documents in decreasing order of the scores. A user can find top documents in the ranking according to their relevance scores.

A particular challenge for SCR using such automatically created transcripts is the presence of recognition errors. Errors in recognised words can impact on retrieval effectiveness, leading to failure to retrieve relevant items where words in a query are missing from the transcript, retrieval at low rank in a retrieved list where there is only a partial match with the query, or retrieval of non relevant items where a mis-recognised word is matched to the query. Larson and Jones (2012) discuss the issue of speech transcripts where word error rates (WERs) higher than 30-40% are found to have significant negative impact on search operations over spoken content archives. Even with lower average WERs, retrieval effectiveness for individual relevant documents can be significantly impacted when matching queries with relevant documents which have been poorly transcribed.

The scale and content diversity of SCR research has expanded over the time. The first large scale SCR research was conducted in the Spoken Document Retrieval Track of the Text REtrieval Conference (TREC) from 1997 to 2000 (Garofolo et al., 2000). The speech corpus used for retrieval contained over 500 hours of broadcast news programmes. The scale and content diversity were increased at the Search and Hyperlinking Task organised at the MediaEval workshop from 2012 to 2014 (Eskevich et al., 2014). The Blip10000 corpus (Schmiedeke et al., 2013) used for the 2012 task contained over 3,000 hours of amateur videos while the BBC corpus used for the 2014 task contained over 4,000 hours of various types of broadcast programmes from BBC. The latest large scale research on SCR has been conducted in TREC Podcast Track in 2020 and 2021 (Jones et al., 2020, 2021). For this study, the podcast corpus contained over 100,000 podcast episodes leading to 50,000 hours of audio.

State-of-the-art ASR systems have been reported to produce very low WERs on established well-defined tasks e.g., transcription of broadcast news and audiobooks (Hadian et al., 2018; Lüscher et al., 2019; Thomas et al., 2019). However, low transcription error rates can be observed only when the domain of the training data and testing data for an ASR system overlaps. The lexical domain mis-match issue could be resolved by developing open-vocabulary ASR systems consisting of sub-word units. While this approach has been successful for the task of finding a keyword in speech (i.e., keyword spotting) or spoken document search that can contain out-of-vocabulary words which ASR systems cannot transcribe (James and Young, 1994; Akiba et al., 2011), our goal is to overcome poor recognition of speech when acoustic domain mis-match occurs. User-generated spoken content is often highly varied in terms of speaker demographics, speaking styles, and acoustic conditions (Schmiedeke et al., 2013). For the volumes of user-generated spoken content available now, ASR systems must be used on content domains for which they have not been trained. While the use of deep neural networks (DNNs) for ASR has significantly improved the quality of speech transcripts in recent years (Hinton et al., 2012), domain mismatch is still observed to produce higher WERs (Narayanan et al., 2018; Moriya and Jones, 2021b).

Standard SCR systems run a retrieval model over transcripts generated using an

ASR system. To mitigate the effects of transcription errors from ASR on retrieval output, the retrieval model can be extended to use N-best ASR hypotheses instead of 1-best hypotheses or use hypotheses consisting of sub-word units in combination with standard hypotheses consisting of words (Larson and Jones, 2012). More recent work on SCR has focused on learning representations from acoustic signals and searching for query terms directly on spoken documents without transcription (Chen et al., 2018). Though there has been interest in effective neural methods in IR (Guo et al., 2020), such methods have so far rarely applied to the SCR task. The SCR challenges lying in diverse multimedia archives have not been properly investigated in existing work.

1.1 Overview of Topics in this Thesis

This thesis is focused on the challenges to effective SCR arising from ASR transcription errors, in particular motivated by the highly varied speech data found in user-generated spoken content archives. In this thesis, we focus on enhancing SCR for diverse data sets via improvement of ASR systems with the aim of generating more accurate speech transcripts of user-generated spoken content which can directly benefit search operations for spoken content, but also augmentation of SCR technique with seek to enable the SCR system to better handle noisy ASR transcripts. This thesis consists of three research investigations focusing on improvement of ASR and SCR systems: (i) use of multimodal data to improve ASR, (ii) development of multi-domain ASR which can better handle highly varied user-generated data, and (iii) augmentation of neural SCR ranking methods for noisy ASR transcripts.

1.1.1 Multimodal augmentation of ASR systems

This investigation is motivated by the fact that human speech understanding exploits situational contexts obtained from visual information (McGurk and MacDonald, 1976; Tanenhaus et al., 1995). In the context of ASR used for SCR, the motivation for the use of visual information is to seek to improve correct recognition of "keywords" for search including proper nouns and entities to match relevant documents with queries from a visual channel. In other words, if such entities and proper nouns are present in both visual and speech channels, the use of visual information can help ASR to accurately transcribe such terms. User-generated videos are accompanied by independent visual and speech channels with user-created metadata including video titles. Similar to visual information, if "keywords" or their synonyms are present in both titles and speech, the use of video titles can help ASR for transcription of nouns and entities. Current ASR systems generally exploit only the speech channel for speech transcription. This investigation explores integration of information from the visual channel and meta-data into the ASR system to seek to improve ASR accuracy.

1.1.2 Transcription of topically diverse speech

When creating automatic transcripts of highly varied user-generated spoken content, an ASR system can be required to transcribe content outside the domain for which it has been trained. Given the volume and diversity of user-generated content available now, it will be unrealistic to prepare manually transcribed training data and build a domain specific ASR system for every speech domain. The goal of this investigation is to examine use of a single ASR system to handle highly varied content better by use of semi-supervised training and acoustic model adaptation. Semi-supervised training for ASR uses an existing ASR system trained on out-of-domain data to transcribe untranscribed data of in-domain data (Veselỳ et al., 2013). A new ASR system can be trained on combination of out-of-domain data accompanied by manual transcripts with in-domain data for which automatic transcripts are created. Acoustic model adaptation can adapt an ASR system to particular acoustic conditions including specific speakers and environmental noise (Abdel-Hamid and Jiang, 2013). For the challenge of transcribing highly varied content, acoustic model adaptation can be used to enable a single ASR system to transcribe multiple data domains by providing the system with information about data domain. While the adapted ASR system is evaluated with WERs, we are interested primarily in impact on its SCR effectiveness.

1.1.3 Augmentation of SCR ranking methods

While there are a number of established IR models which have been applied successfully to SCR, there has recently been active research on neural network based document ranking methods, so called *neural ranking* and *BERT ranking* (Guo et al., 2020; Lin et al., 2020). Little work investigating the use of these methods for SCR has been reported. This investigation explores whether neural ranking and BERT ranking models with erroneous ASR transcripts for SCR and potential extension of neural ranking and BERT ranking models for SCR with ASR systems. In this thesis, the use of N-best ASR hypotheses for the neural ranking method is investigated to overcome retrieval errors caused by noisy ASR transcripts.

1.2 Contributions

As mentioned above, SCR systems generally exploit ASR transcripts to search for relevant spoken documents given a user query. Despite recent improvement of ASR systems, recognition accuracy can drop when ASR systems are applied to highly diverse data (Moriya and Jones, 2021b). This motivates SCR research to be conducted from two perspectives: (i) improvement of ASR systems for highly diverse content for SCR and (ii) augmentation of ranking models to overcome ASR transcription errors. Contributions of research investigations conducted in this thesis are as follows:

- Investigations of multimodal features including visual context features, video title features and speaker face features for ASR systems.
- The use of semi-supervised methods for improvement of ASR systems for highly diverse spoken content.

- Investigations of the effects of semi-supervised ASR transcripts on SCR.
- The use of N-best ASR transcripts to improve robustness of ranking methods for ASR transcription errors.

1.3 Thesis Structure

The ever growing online archives of diverse spoken content is increasing the demands for effective SCR systems. Standard SCR systems generally exploit automatic speech transcripts created using ASR. Errors in ASR transcripts can affect SCR effectiveness. While research in ASR generally focuses on reducing word error rate, in this work we are interested in improving ASR for SCR[1]. Similarly, while standard IR methods have been shown to be relatively robust to errors in the transcript, there has been little work exploring the development of IR methods specifically for noisy ASR transcripts. These topics will be explored in this thesis. The structure of this thesis is organised as follows.

Chapter 2 provides the background technologies used for ASR. The chapter begins with an overview of the standard architecture of ASR systems focused on the state-of-the art hybrid hidden-Markov model (HMM) deep neural network (DNN) model used for ASR. Following the overview on ASR, visual augmentation of ASR, semi-supervised training for ASR and acoustic model adaptation are discussed. Research questions relevant to ASR used for SCR are presented at the end of this chapter.

Chapter 3 provides an overview of the standard SCR pipeline. Following the SCR overview, the standard BM25 probabilistic retrieval model and a description of recently proposed neural ranking and BERT ranking models are presented. Research questions relevant to retrieval models are proposed at the end of this chapter.

Chapter 4 overviews existing research on SCR and its history. As mentioned ear-

[1] The proposed methods of this thesis to improve ASR systems could be useful for general-purpose ASR

lier, the first large scale study of SCR was the TREC Spoken Document Retrieval Track using broadcast news programmes (Garofolo et al., 2000). This chapter summarises SCR research following the initial TREC task to provide readers with the background contexts of the research carried out in this thesis.

Chapter 5 describes our experimental investigation into the integration of visual information into the ASR system. The chapter demonstrates integration of visual features into the ASR system. Experiments are conducted to examine the use of visual context features and visual human faces for augmentation of the ASR system.

Chapter 6 presents our exploration of semi-supervised training for ASR and acoustic model adaptation using multimedia content genre. Experiments using these techniques are evaluated in terms of WERs.

Chapter 7 examines the effectiveness of our semi-supervised ASR transcripts for SCR and examines the use of neural ranking and BERT ranking models for SCR and their extension to allow for ASR errors. Experimental results are reported to show that the neural ranking system can be useful for the SCR task and the proposed approach to extending neural ranking models can enhance retrieval of spoken user-generated content.

Chapter 8 concludes the research findings of the thesis. The chapter reviews the research questions proposed against experimental results. The chapter ends with suggestions for further investigations.

Chapter 2

Automatic Speech Recognition

This chapter provides a technical overview of standard state-of-the-art ASR methods, and their limitations with respect to ASR. The chapter begins with an explanation of the pipeline architecture of ASR consisting of acoustic feature extraction, acoustic modelling, language modelling and decoding. Following the ASR overview, the use of multimodal features for ASR is introduced. This discussion includes extraction of visual features from videos. For the purpose of retrieval of highly varied spoken content, ASR systems need to handle multiple domains. This contrasts with such existing ASR process where the focus is on well designed transcription tasks. A technology with potential to improve ASR for diverse domains is semi-supervised training. The chapter then introduces semi-supervised training and outlines methods for acoustic model adaptation for ASR with the focus on development of multi-domain ASR. The chapter ends with the proposal of two research questions relating to ASR for SCR applications.

2.1 ASR Overview

The fundamental goal of the ASR process is to produce the most accurate transcript of a speech input by calculating the most likely word sequence $\hat{W}$ given the input

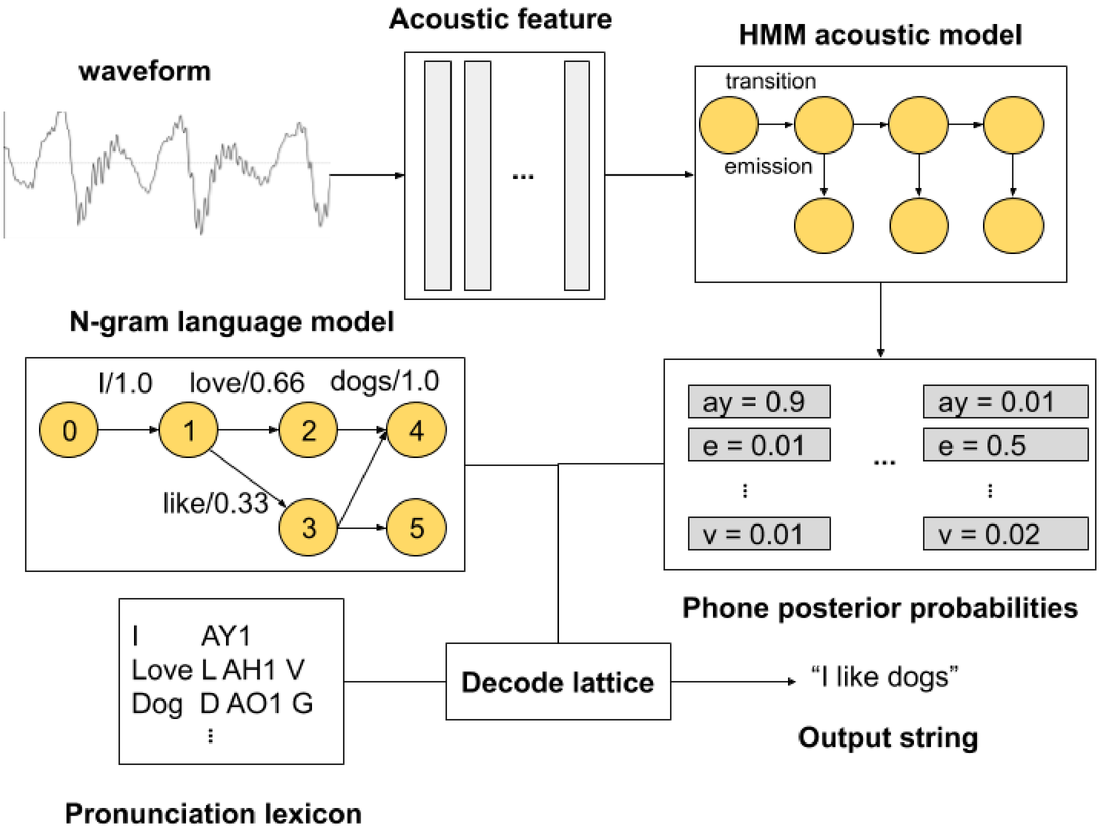


Figure 2.1: Schematic diagram of a HMM based ASR architecture. The HMM acoustic model and the N-gram language model are trained offline.

acoustic signal. This can be formally expressed as in Equation 2.1.

$$\hat{W} = \arg\max_{W \in L} P(O|W)P(W) \tag{2.1}$$

where O is an observed acoustic signal, L is a vocabulary, $P(O|W)$ is the likelihood of the observed acoustic signal relating to a spoken word sequence, and $P(W)$ is the probability of a given word sequence. In this process, an input audio signal is a noisy representation of an original message and an output word sequence is the denoised version of the message.

Figure 2.1 shows the schematic diagram of a classic ASR system consisting of three components: acoustic model, language model and pronunciation lexicon. The input to the ASR system is a sequence of acoustic feature vectors extracted from an input speech audio waveform (Section 2.1.1). The acoustic model takes as input a sequence of acoustic feature vectors and produces a sequence of sub-word phone posterior probabilities. The acoustic model of the traditional standard ASR uses HMMs which are suitable for modelling the sequential nature of speech (Jelinek, 1976; Gales and Young, 2008). Each state of the HMM stores a probabilistic model for prediction of a pre-defined acoustic phone label given an acoustic feature vector. For the choice of probabilistic model, in recent years, DNNs have demonstrated superior recognition accuracy to conventional Gaussian mixture models (GMMs) (Hinton et al., 2012). More details of acoustic modelling are given in Section 2.1.2.

The language model is trained on word sequences contained in training data and the model supports the reconstruction of word sequences likely to appear in spoken data. Within the traditional standard ASR architecture, the language model is built with N-grams and with the N-gram language model converted to form a decoding graph (Mohri et al., 2002). Similar to acoustic modelling, neural methods have become popular for language models. While a neural model for the acoustic model is directly incorporated into HMMs, neural methods for language models is often employed as a post-processing stage after decoding in the form of lattice re-scoring or ASR N-best re-scoring (Mikolov et al., 2010). More details of the N-gram and neural language models are presented in Section 2.1.3.

The process to choose the most likely word sequence is referred to as *decoding* (Section 2.1.4). A decoding graph is created by integrating sub-word phone paths of the HMM acoustic model into word paths of the N-gram language model by using the pronunciation lexicon which keeps word tokens and corresponding phone sequences. The decoding process searches through the graph for the path with the highest likelihood. This path corresponds to the most likely word sequence, and is the transcript output by the ASR system.

2.1.1 Acoustic feature extraction

The goal of acoustic feature extraction is to produce an audio representation which can be used with an acoustic model to distinguish the differences between phones.

Any digitised sounds are characterised by a time series of values that can be drawn as a sound waveform by sampling the analogue speech signal. The frequency spectrum of a speech waveform can be considered stable within a short time region. The speech series of samples are thus segmented into short time regions referred to as frames which are assumed to contain a stable speech waveform. However, speech frames may miss significant transitions from one phone to another. To avoid this problem, speech frames are often overlapped. For example, a speech frame of 25 milliseconds can be overlapped by 10 milliseconds between frames. A Fourier transform is then applied to each speech frame to represent the acoustic information in terms of the energy in a range of frequency bands. For input of the neural acoustic model, the widely used acoustic feature is the filter bank feature (Mohamed et al., 2012). To generate the filter bank feature, a Mel-filter is applied to speech frames after Fourier transform (Davis and Mermelstein, 1980). The Mel-filter mimics human perception of sounds which is more sensitive to lower frequency. Another important acoustic feature to mention is Mel-frequency cepstral coefficients (MFCCs) (Davis and Mermelstein, 1980) which are created by applying discrete cosine transform to the filter bank feature. The advantage of the MFCCs is that the feature is not correlated and is easy to compress.

2.1.2 Acoustic modelling

The acoustic model takes as input a sequence of extracted acoustic feature vectors and outputs phone posterior probabilities that are later used in the decoding step described in Section 2.1.4. This section introduces a standard ASR acoustic model using an HMM and incorporation of neural models into HMMs as used in state-of-the-art ASR systems.

Hidden Markov Model

A Hidden Markov model is a graphical model suitable for modelling sequential data. The model assumes a Markov process, where prediction of the current phenomenon

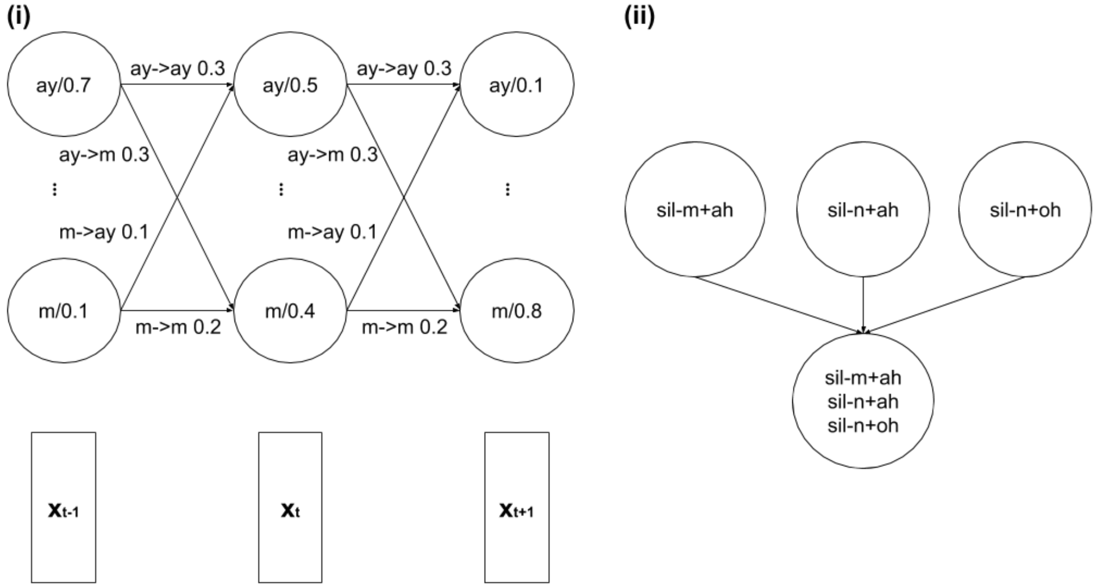


Figure 2.2: **(i)** Graphical representation of an acoustic HMM model, and **(ii)** context-dependent phone modeling and state tying. State tying is a technique to model acoustically similar context dependent phones to share a single state in order to reduce the number of phones which need to be modelled.

can be based only on the current and the one previous observation. Since speech data has a sequential property, HMMs are widely used to convert a sequence of acoustic feature vectors to phone probabilities (Jelinek, 1976; Gales and Young, 2008). An acoustic HMM is shown in Figure 2.2. HMMs consist of: (i) a set of states, (ii) a transition probability matrix, and (iii) emission probabilities. In sub-word speech modelling, each HMM consisting of three to five states to represent a phone, although one state represents a phone in Figure 2.2 for simplicity. The transition probability matrix stores information about how likely the model is to move from state i to state j as each acoustic vector is input. Emission probabilities express how likely state i is to generate a feature vector at time t. To generate emission probabilities, each state in an HMM stores a probabilistic model. A GMM was conventionally a popular choice for modelling emission probabilities. However, it has been demonstrated that incorporating a DNN into an HMM produces significantly higher recognition accuracy (Hinton et al., 2012), and this has become established as the standard approach to acoustic modelling in ASR. In Figure 2.2 (i), there are two phones "ay" and "m" forming a token "I'm".

Context dependency and state tying

The acoustic model illustrated in Figure 2.2 (i) is context independent, where each state represents a single phone label. An established technique to improve phone recognition accuracy is context dependent phone modelling (Lee, 1990; Young et al., 1994). This is motivated by the fact that acoustic properties of a phone can be different depending on the position of the phone in a word (e.g., l as in "latin" and "hall") and the phones which precede and follow it. Context dependent phone modelling, however, exponentially increases the number of output classes to model. The increased number of output classes causes inefficiency in decoding due to the increased number of states in the HMM, and requires more training data to include enough acoustic examples for each context dependent phone. In order to resolve these issues and to reduce the number of context dependent phone labels, acoustically similar phones (e.g., nasals "m" and "n") can share the same states (Lee, 1990). Alternatively, a decision tree can automatically determine context-dependent phone labels which share the state for computation of emission probabilities (Young et al., 1994). These techniques to reduce the total number of context-dependent phone labels are referred to as state tying. Tied-states are responsible for emission probabilities of two or more context-dependent phone labels. Figure 2.2 (ii) shows an example of context-dependent phone modelling and state tying. In the figure, the left most state is a phone “m” preceded by silence and followed by “ah” (sil-m+ah). Since this is phonetically similar to the other two states, sil-n+ah and sil-n+oh, these three states can be modelled as the same phone state to reduce the number of context-dependent phones to model. ASR systems investigated in this thesis estimate context dependent phone probabilities rather than single phones and use state-tying to reduce the number of output phone labels.

Gaussian Mixture Model

Until the early 2010, a GMM was a popular choice to compute an emission probability of an HMM state. A GMM attempts to model acoustic characteristics of each phone by multiple Gaussian components interpolated with mixture weights. A GMM computes a probability of a phone being generated from a GMM with mean and covariance of all Gaussian components μ_*, $\mathbf{\Sigma}_*$ and mixture weights $\mathbf{c}_*$ which satisfies $\sum_{k=1}^{M} c_k = 1$, given a speech feature vector $\mathbf{x_t}$, as defined in Equation 2.2

$$P(\mathbf{x_t}|\mathbf{\Sigma}_*, \mu_*, \mathbf{c}_*) = \sum_{k=1}^{M} c_k \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_k|^{\frac{1}{2}}} \exp[-\frac{1}{2}(\mathbf{x_t} - \mu_k)^\top \Sigma_k^{-1}(\mathbf{x_t} - \mu_k)] \tag{2.2}$$

where M is the number of mixture components and D is the vector size of each data point.

Deep neural network

Since 2012, a DNN has become a common choice of a probabilistic model to be incorporated into an acoustic HMM by its demonstrated improvement of recognition accuracy over the traditional GMM (Hinton et al., 2012). A DNN consists of a number of weights interconnected between nodes, and a layer referred to as a collection of nodes interconnected with weights (edges). Hinton and S. Osindero (2006) named an artificial neural network with 3 layers a deep network. Suppose a neural network has a single layer and an audio feature vector is $\mathbf{x}$, a probability of a phone being j can be computed as follows:

$$\begin{aligned} \mathbf{h} &= \sigma(\mathbf{W} \cdot \mathbf{x} + \mathbf{b}) && (2.3) \\ p(y = j|\mathbf{x}) &= Softmax(h_j) = \frac{exp(h_j)}{\sum_{l=1}^{K} exp(h_l)} && (2.4) \end{aligned}$$

where $\mathbf{h}$ is a hidden representation of input $\mathbf{x}$, $\mathbf{W}$ is a weight matrix of a network, $\mathbf{b}$ is a bias term to prevent a network from overfitting in training data, σ is a non-linear activation function such as sigmoid, and h_j is the jth value of a hidden representation $\mathbf{h}$.

An advantage of DNNs over GMMs is that “deep” architecture forms a complex function capable of transforming audio features into an abstract representation and translating this to phone probabilities. DNNs are also better at modelling correlated features than GMMs. Filter bank features introduced in Section 2.1.1 are highly correlated and GMMs are required to have many diagonal covariance matrices or full covariance matrices to model correlation (Hinton et al., 2012). DNNs, on the other hand, require a larger amount of training data for acoustic modelling than GMMs.

DNN models are trained using the "back-propagation" algorithm (Rumelhart et al., 1986). In the context of DNN models for acoustic modelling, suppose each acoustic feature vector is associated with one phone label, the model predicts the probabilities of all phone labels using Equation 2.4. Since the target labels are available for each input acoustic feature, the total error of model predictions can be computed using the cross-entropy shown in Equation 2.5.

$$E(\mathbf{y}|\hat{\mathbf{y}}) = -\sum_{l=1}^{K} y_l \log \hat{y}_l \tag{2.5}$$

where $E(\mathbf{y}|\hat{\mathbf{y}})$ is the total error of model predictions given predicted labels, y_l is a value of lth output class of label and $\hat{y}_l$ is a predicted probability of lth output class. The back-propagation algorithm attempts to reduce the total error of model predictions by adjusting weights and weight adjustment is propagated from top layers to bottom layers backwards.

2.1.3 Language modelling

The language model (LM) in an ASR system estimates the likelihood of a sequence of words within the language. Language models in ASR systems are conventionally built with N-grams, since N-grams retain a convenient format to be converted to a decoding graph combined with an acoustic HMM topology. Recently, a DNN-based language model, often referred to as a neural language model, has been used for language modelling and shown significant improvement of performance in terms of recognition accuracy (Mikolov et al., 2010). This section introduces both conventional N-gram language models and the currently used neural language models.

Standard N-gram Language Model

An N-gram language model produces a probability of a word given the $N-1$ previous context words. For instance, a bigram model takes one previous word to compute a probability of the current word and a trigram model takes two previous words. The probability of a word sequence can be estimated by taking a product of probabilities of the words in the sequence. The N-gram language model is built using maximum likelihood estimate (MLE), that computes a maximum probability of a target word following N context words. The MLE for N-gram language modelling can be defined as in Equation 2.6.

$$P(w_n|w_{n-N+1:n-1}) = \frac{C(w_{n-N+1:n-1}w_n)}{C(w_{n-N+1:n-1})} \tag{2.6}$$

where a N-gram probability of the word w_n is computed by dividing the number of times w_n following $N-1$ words $w_{n-N+1:n-1}$ observed by the number of times $w_{n-N+1:n-1}$ observed in a training corpus. While increasing the context size can generalise the model, doing this requires more space to store word probabilities.

The most widely used metric to evaluate a language model is perplexity which computes how well a model can predict words in an unseen test corpus. Perplexity can be computed as shonw in Equation 2.7.

$$PP(W) = N\sqrt{\frac{1}{P(w_1 w_2 ... w_N)}} \tag{2.7}$$

where $P(w_1 w_2 ... w_N)$ is a product of probabilities of the words in the sequence. A model is a better fit in an unseen corpus, when it produces lower perplexity.

When building an N-gram model, a pre-defined vocabulary is often used to avoid estimating probabilities of infrequent words and words outside the vocabulary are seen as an unknown word. In other words, all of the words existing in the world cannot be covered by the model and such words are out-of-vocabulary (OOV) words. An OOV issue arises when words are uttered in speech but not modelled by a language model. Such words have a zero possibility of being correctly transcribed.

Smoothing is used to alleviate the inevitable issues of sparsity of training data for N-grams with infrequent words and word sequences in the training data. Smoothing takes some probabilities from N-grams of higher counts and assigns them to those of lower counts. The most widely used smoothing is interpolated Kneser-Ney smoothing (Chen and Goodman, 1995). For example, when a training corpus contains many instances of "San Francisco", both a bigram count and a unigram count of "Francisco" becomes high, even though "Francisco" can appear only when preceded by "San". Kneser-Ney smoothing penalises probabilities of such words which appear only in specific contexts.

Neural Language Model

An alternative to the N-gram language model is a neural language model, specifically one using a recurrent neural network (RNN) (Mikolov et al., 2010). It has been demonstrated that neural language models in ASR systems produce better recognition accuracy than the N-gram language model (Mikolov et al., 2010). The RNN is a special version of the DNN, where not only layers but also an input history contributes to computation of abstract representation of input. While the N-gram language model can consider only N context words, a neural language model using RNNs can take account of a longer context history. Given an input word converted into a vector representation $\mathbf{x_t}$, a probability of the next word being j can be computed as follows:

$$\mathbf{h_t} = \sigma(\mathbf{U} \cdot \mathbf{x_t} + \mathbf{W} \cdot \mathbf{h_{t-1}}) \tag{2.8}$$

$$p(y = j | w_t, ..., w_1) = Softmax(h_t^j) \tag{2.9}$$

where $\mathbf{U}$ and $\mathbf{W}$ are weight matrices for an input vector $\mathbf{x_t}$ and a context history $\mathbf{h_{t-1}}$, $\mathbf{h_t}$ is a hidden representation of the current input, and h_t^j is jth value of a hidden representation $\mathbf{h_t}$. Since the context history $\mathbf{h_{t-1}}$ is propagated from the first input $\mathbf{x_1}$, the model is aware of all preceding input unlike the N-gram language model. The RNN is often employed with a long-short term memory (LSTM) cell that keeps a cell state in addition to an input history (Hochreiter and Schmidhuber, 1997). A LSTM cell is an extension of a RNN that incorporates additional internal representation units to more efficiently learn long-term contexts. LSTM cells were developed to deal with a vanishing or exploding gradient problem, where during training of an RNN language model, errors between model prediction and labels are multiplied through a long input sequence and cause a numeric underflow or overflow. The LSTM cell state decides to retain and discard information from a history using the gates, and can efficiently train a model without causing such an issue.

2.1.4 Decoding

Decoding is the process of searching for the most likely word sequence through a large weighted finite-state transducer (WFST) called a decoding graph (Mohri et al., 2002). Modern ASR systems use a decoding graph constructed by combining the output context-dependent phone labels of the acoustic model, translation of phone labels into words stored in the pronunciation lexicon and the N-gram language model. Figure 2.3 shows an example WFST where the language model and the acoustic HMM are combined. For example, the edge from node 1 to node 4 is associated with "like" with a probability 0.4. This path is obtained from an entry in the pronunciation lexicon mapping "like" from a sequence of phones "l ay k". In Figure 2.3, four different sentences can be produced: "I {like,love} {me,moo}".

A decoding algorithm searches through the decoding graph and finds the path which has the highest score (Viterbi, 1967). However, due to the large size of the graph, considering all possible paths is prohibitively expensive. A beam search method is employed to reduce the number of hypotheses to consider while searching through a graph. Instead of producing the best path from the decoding graph, the output of decoding can be stored in a lattice format which is a representation of reasonable alternative hypotheses. The lattice retains phone posteriors and language model scores (Mohri et al., 2002). Re-scoring of acoustic and language scores using complex models without heavy computation cost can be applied to the lattice, and

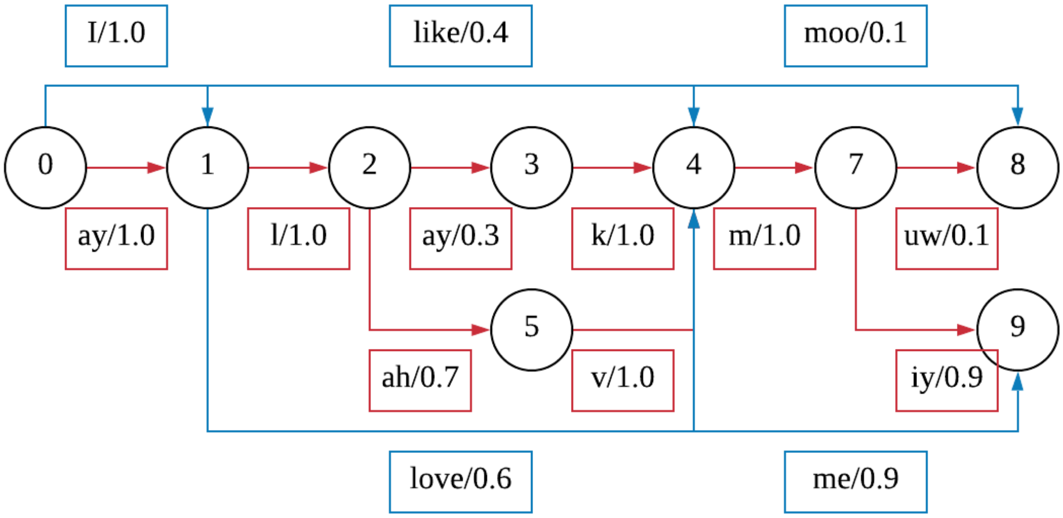


Figure 2.3: Example WFST combining an N-gram language model and an acoustic HMM. Each edge is associated with a phone label or a word and its probability. Blue edges show transitions of the language model, while red edges are associated with transitions of phones.

finding decoded transcripts from the lattice generally have improved accuracy.

Unlike the N-gram language model, the RNN language model introduced in Section 2.1.3 is not feasible to incorporate into the decoding graph due to its dependency on long-term context. The RNN language model is therefore employed in lattice-rescoring (Liu et al., 2014; Xu et al., 2018) or N-best re-scoring (Mikolov et al., 2010), in which the top N scoring transcript hypotheses are decoded from the lattice prior to re-scoring. In N-best re-scoring, suppose an ASR hypothesis is h, the score of this hypothesis from the RNN language model $L_{RNN}(h)$ can be computed in Equation 2.10.

$$L_{RNN}(h) = -\sum_{i=1}^{I} logP(x_i|x_1...x_{i-1}) \tag{2.10}$$

where I is the number of words in the hypothesis. The total score of the hypothesis is interpolated with the acoustic model score as in Equation 2.11.

$$L(h) = \lambda * AM(h) + (1-\lambda) * L_{RNN}(h) \tag{2.11}$$

where $AM(h)$ is the score of the hypothesis from an acoustic model and λ is a weight to determine the importance of both scores.

2.1.5 Word error rate

ASR hypotheses are commonly evaluated using the word error rate (WER) metric. Transcription errors are classified into three types: deletion, insertion and substitution. Deletion is an error where a word in a reference is not transcribed in an ASR hypothesis text i.e., the word is missing from the transcript, while insertion is an error where a word not present in a reference text is added to an ASR hypothesis text. Substitution occurs when a word in a reference text is replaced for another word in a hypothesis text. While forcing an ASR model to produce fewer words reduces insertion errors, this will increase the number of deletion errors and vice versa.

To determine the number of deletion, insertion and substitution errors in an ASR transcript, an optimal alignment of a reference word sequence with a hypothesis word sequence is computed using a dynamic programming algorithm. The WER of an ASR hypothesis is computed as shown in Equation 2.12.

$$WER = \frac{N_{del} + N_{ins} + N_{sub}}{N_{total}} \tag{2.12}$$

where N_{del} is the number of deleted words, N_{ins} the number of inserted words, N_{sub} the number of substituted words and N_{total} the number of total words in a reference.

In the context of SCR, improvement of the average WER of ASR hypotheses generally leads to better search effectiveness. As discussed in Section 1, Larson and Jones (2012) review existing research on SCR and search effectiveness can generally degrade when WER is higher than 30-40%. Garofolo et al. (2000) point out that there was a near-linear relationship between search effectiveness and WERs. Johnson et al. (1999b) also observed a fall in search effectiveness with WER higher than 30%. Similar observations on the relationship between search effectiveness and WERs are also reported for the task of lecture video retrieval (Chelba et al., 2007) and call center speech (Mamou et al., 2006).

Although alternative evaluation metrics for ASR with respect to SCR can be considered, a WER is still a reasonable metric to employ for its simplicity and somewhat high correlation with search effectiveness. Garofolo et al. (1998) explored two evaluation metrics that were hypothetically more suitable for ASR evaluation with the SCR purpose. Their *stop-word-filtered* WER removes common words and *stemmed stop-word-filtered* WER applies stemming of words in addition to stop word removal before ASR evaluation. These metrics did not show much higher correlation with search effectiveness than WERs. Johnson et al. (1999b) also proposed a term error rate (TER) that is independent of word order unlike WERs, but they did not conclude that TER was a better ASR metric than WER with respect to SCR. The only alternative ASR evaluation metric that showed a higher correlation with search effectiveness than WER was a named entity word error rate proposed by Garofolo et al. (1998). The correlation coefficient of WER with search effectiveness was .85 and the named entity word error rate increased the coefficient to .91. However, this evaluation metric requires manual annotation of named entities in ASR reference transcripts and has not been employed in other literature.

In this section, the WER is introduced as an evaluation metric of ASR systems. Perplexity is another metric that evaluates a language model incorporated into an ASR system (Section 2.1.3). Other metrics to evaluate search effectiveness including mean reciprocal rank (MRR), mean average precision (MAP) and normalised discounted cumulative gain (nDCG) are introduced in Section 3.4.

2.2 Multimodality and ASR

Standard ASR methods use only the audio speech signal. However, human speech and language processing is inherently a multimodal process. Humans seamlessly update situational and environmental contexts, and their interpretation of language can depend on these contexts. Psycholinguistics research shows that language interpretation in the real world is a multimodal process (Tanenhaus et al., 1995). Participants in this experiment were presented with a picture of an apple on a towel, a towel without an apple and a box. When the sentence "put the apple on the towel" was played, their gaze moved to the towel without an apple before the full sentence finished. The full sentence in this experiment was "put the apple on the towel in the box". This demonstrates that participants predicted upcoming words based on the situational contexts presented in the picture. The McGurk effect is another example of visual information affecting human interpretation of speech (McGurk and MacDonald, 1976). When participants were presented with a video of lip movement producing a labio-dental phone (e.g., "v") and a bilabial voiceless stop "b" was played, they were not able to perceive the sound of "b" correctly. These results from behavioral psychology motivate us to explore the use of multimodal information for ASR processes.

Audio-visual speech recognition (AVSR), motivated by the McGurk effect, has aimed to improve recognition accuracy by integrating lip movement into ASR. AVSR techniques can be used in a hybrid HMM-GMM ASR system (Potamianos et al., 2003) and for a HMM-DNN system (Ngiam et al., 2011). AVSR has been demonstrated to show enhanced recognition accuracy in noisy conditions. However, application of AVSR has been limited due to the difficulties in constructing audio-visual data in which speaker frontal face is always present and lip movement can be seen. The goal of our investigation of multimodal ASR in this thesis is to enhance ASR and SCR for unconventional spoken video content not restricted to recognising the speech of an individual talking directly to a camera. Videos in general can be highly varied in their visual patterns and image quality may be too low to extract fine-grained features such as precise lip movements.

Other work on multimodal ASR has aimed to integrate not only lip movement but also any background contexts of speech available in a visual stream or meta-data of spoken content. Specifically, this topic focuses on integration of information taken from a visual stream or meta-data of multimedia content. Previous work has explored event patterns seen in a visual stream of baseball games (Fleischman and Roy, 2008) to mitigate audio noise included in baseball commentary, spatio-temporal information of conversation (i.e., when and where conversation happened) that was useful to predict certain word (e.g., coffee) uttered at a certain time or a place (e.g., morning at a living room) (Roy et al., 2014), speaker attributes extracted from a speaker face (i.e., age, gender, and race) (Miao et al., 2014), object, scene and action information taken from a visual stream (Gupta et al., 2017). It has been shown that all of the multimodal features above contributed to reduction of WER. An interesting connection between background information and speech was a scene feature extracted from a video which mitigated background noise (e.g., noise from cars), when a speaker was outside a building. Notably, work from Gupta et al. (2017) has been extended to the end-to-end ASR framework (Caglayan et al., 2019; Palaskar et al., 2018). Along with the line of existing work on multimodal ASR, this thesis investigates visual context features and textual features (i.e., video titles) for RNN-LM adaptation and speaker face features for acoustic model adaptation for improvement of the WER.

2.2.1 Visual feature extraction

Visual features can be extracted using a computer vision technique and such features can be used to achieve multimodal adaptation of ASR. In computer vision, there has been significant progress of image recognition in recent years using deep learning. State-of-the-art image recognition models use many convolutional layers (He et al., 2016). Regardless of the image recognition model architecture, computer vision models pre-trained on a large scale dataset such as object recognition datasets and face recognition datasets are often made available to public (Russakovsky et al., 2015; Parkhi et al., 2015). Such models can recognise objects in a given image when trained on images with labelled object names or can recognise faces of high profile figures when trained on a dataset containing images of faces with their name labels. An important concept of pre-trained image recognition models is that these

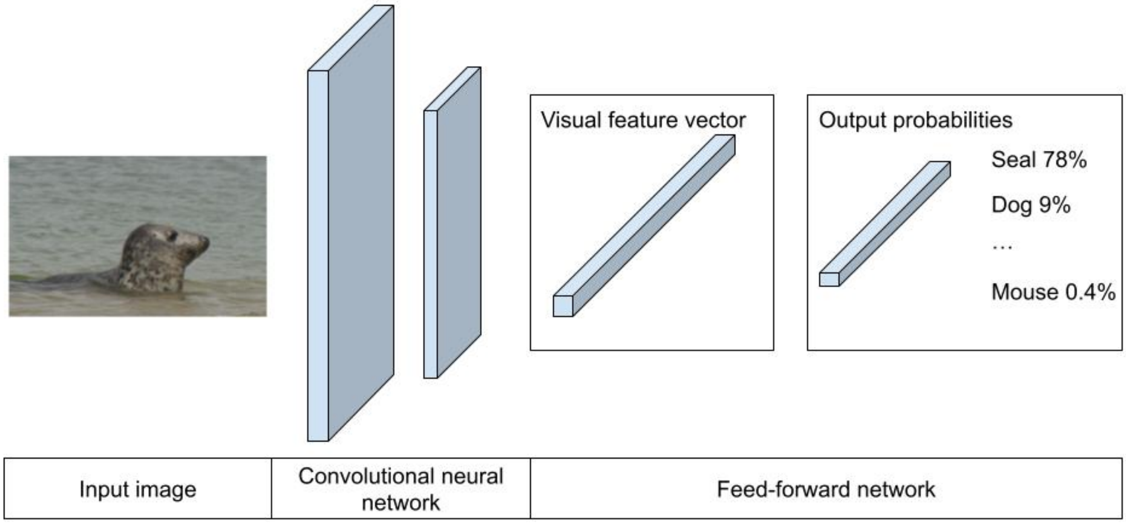


Figure 2.4: Example process to extract a visual feature using an animal recognition model.

models can produce an abstract representation of a given image and can be used as a visual feature extractor. Figure 2.4 illustrates an example process to extract a visual feature representation using the model that can recognise animals. The input image goes through several convolutional operations and these operations produce 3D tensor features. The output of the covolutional operations is a feature vector which is an input of a feed-forward network. The final layer of the model is an output classification layer which returns probabilities of classes defined in a dataset. Output from the penultimate layer of the model is taken as a visual feature of fixed length embedding vector of an image. In this example, the extracted feature is high level visual representation of an animal given an input image.

This thesis investigates the use of visual context features and speaker face features for ASR. The visual context features representing background contexts of speech present in a visual stream can be extracted from a pre-trained image recognition model, hypothesising that the pre-trained model retains abstract knowledge of images. For example, when a speaker in a video holds a tennis racket and stands on a tennis court, the topic most likely covers tennis and such information can be inferred from the visual context. Similarly, a face recognition model can be used as a feature extractor of speaker face features. The main advantage of using these pre-trained models is that it is not necessary to build a new model specially for the ASR task.

2.3 Multi-domain ASR

ASR systems are generally developed to work most effectively for speech of a particular topical domain. These systems can perform poorly on data outside the domain on which they have been trained. It has been demonstrated that the latest ASR systems can achieve WERs of as low as 5% (Hadian et al., 2018; Lüscher et al., 2019). However, this is limited to the laboratory settings where the systems are trained and evaluated on the same domain data. The domain is typically audio-books and broadcast news which is free of background noise. On the other hand, Moriya and Jones (2021b) show that WERs could be as high as 40% when those systems are applied to spoken content with spontaneous conversations and background noise such as street interviews and game commentaries. By contrast, multi-domain ASR aims to recognise speech from multiple topical domains. The challenges for multi-domain ASR are accurate transcription of speech from the highly varied speaker characteristics (adult, child and non-native speaker), speaking styles (scripted, formal and informal interviews, sports and video game commentary, and casual conversations), and acoustic conditions (background music, loud audience, applause, and street noise) (Moriya and Jones, 2021b). Multi-domain ASR is important for search applications where speech collections to be searched can be highly and unpredictably varied in topic and style.

Multi-domain ASR can be developed by collecting a large amount of manually transcribed data from varied sources and domains. For example, Narayanan et al. (2018) prepared 162,000 hours of manually transcribed data from Voice Search, Dictation, Far-field speech, Call centre and YouTube domains. Their experiments showed improvement of WERs when ASR systems use transcripts from multiple domains. For example, the ASR system trained on voice-search data improved

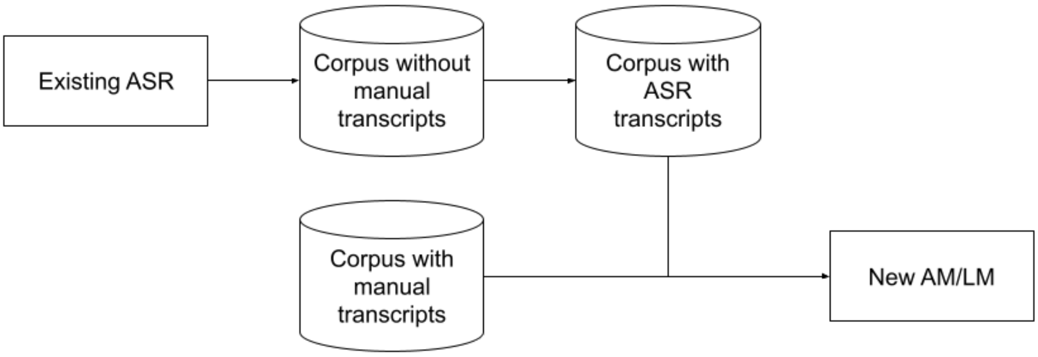

Existing ASR
Corpus without manual transcripts
Corpus with ASR transcripts
Corpus with manual transcripts
New AM/LM

Figure 2.5: Diagram showing the process of semi-supervised training for ASR systems.

recognition of the Telephony test set from WER 31.4% to 20.9% by including transcripts of multiple domains in train data. Nevertheless, curating a collection of over 162,000 hours of manually transcribed data is very expensive and unrealistic in most settings, and creation of manual transcripts can have issues with copyright, making manual transcription difficult. Even with a very large and diverse training collection, the data to be transcribed for search can be outside of the domains covered by the training set. Coverage of such contexts is important for transcript in SCR where a transcription of words related to specific domains is very important.

2.3.1 Semi-supervised training

Semi-supervised training in ASR can be used to alleviate the lack of manual transcripts for highly varied spoken content. A simple process of semi-supervised training for ASR is illustrated in Figure 2.5. Semi-supervised training exploits an existing ASR system to generate a 1-best transcript or decoder lattices of the data for which manual transcripts are not available (Manohar et al., 2018; Veselỳ et al., 2013). A new multi-domain ASR system can then be trained using a combination of the available manual transcripts with transcripts of untranscribed data created using ASR 1-best output or decoding ASR lattices. Semi-supervised training for ASR can be applied to both acoustic modelling (Manohar et al., 2018; Veselỳ et al., 2013) and to language modelling (Çelebi and Saraçlar, 2013; Oba et al., 2013).

Semi-supervised training for a hybrid HMM-DMM system can be significantly improved by using decoded lattices of utterances without manual transcripts rather than 1-best automatic transcripts of utterances (Manohar et al., 2018). Their proposed semi-supervised lattice-free maximum mutual information training is a discriminative training method where the training objective is to predict a sequence of phone labels as a whole, instead of individual phones in an utterance. With this method, several paths of a decoded lattice of utterances without manual transcripts are considered to be the target labels. In other words, when an existing ASR system used for decoding is not confident with several paths of the decoded lattice, these paths will have less impact on training of a new acoustic model.

A large volume of work on semi-supervised training for ASR has developed in recent years. Semi-supervised training has been applied to telephone conversations (Manohar et al., 2018), conversation speech and human-machine dialogue (Sheikh et al., 2020), short message dictation (Huang et al., 2013b), and low data resource scenarios (Carmantini et al., 2019; Su and Xu, 2015; Veselỳ et al., 2013). There is no general consensus on the improvement in WER brought about by semi-supervised training, most likely because methods, corpus domains and the quality of seed systems are so varied. Examining the above work, semi-supervised training tends to bring about 1-2% WER gain at minimum and 7-8% when improvement is large. It should be noted that the semi-supervised method is not equal to lightly-supervised training where partial manual transcripts are available for multiple domains of broadcast programmes (Bell et al., 2015) or various data sources including voice search, dictation and YouTube videos (Narayanan et al., 2018) where improvement gain can be as large as around 10%.

As illustrated above, a number of research projects have examined semi-supervised training for ASR. None of the previous work has focused on our application domain of diverse uncontrolled speech content where manual transcripts are absent for search applications. This thesis investigates whether the use of semi-supervised training for ASR improves WERs on diverse uncontrolled speech, and whether changes in ASR transcripts brought by the semi-supervised method improve search effectiveness.

2.3.2 Acoustic model adaptation

Another technique often employed to mitigate the lack of manual transcripts is acoustic model adaptation. The goal of acoustic model adaptation is to enable a single acoustic model to adapt to multiple data domains. Previous work on acoustic model adaptation considered: speaker specific information (Abdel-Hamid and Jiang, 2013; Saon et al., 2013) and acoustic environmental conditions (Fainberg et al., 2017; Feng et al., 2017; Kim et al., 2016).

A widely used method for adaptation of a neural acoustic model is feeding utterance specific information known as "i-vector" to the acoustic model (Dehak et al., 2011). The "i-vector" is a fixed-dimensional vector feature representing characteristics of a speaker. The "i-vector" was originally proposed for speaker identification, while the feature is also widely used for ASR (Saon et al., 2013). The i-vector extraction process is to build an i-vector extractor trained using factor analysis to compute the latent factor of the utterance of one speaker in comparison to the accumulated features of all speakers.

Since a collection of highly varied spoken content can contain various speaker characteristics, speaking styles and acoustic conditions, acoustic model adaptation is an attractive option to deal with speech of multiple domains. Along with semi-supervised training for ASR, we investigate whether acoustic model adaptation can contribute to the effectiveness of an ASR system in multiple domains. We are aware that adaptation techniques also exist for the N-gram language models (Roy et al., 2014) and for the RNN language models (Mikolov and Zweig, 2012). However, these methods require a sufficient amount of text data to represent each domain class and are applied to manually created texts, while the semi-supervised method only generate texts with errors. Due to these reasons, language model adaptation is not examined in this thesis.

2.4 Research questions

In this section, research questions relevant to ASR investigations are proposed. This chapter has reviewed motivation to integrate multimodal information into ASR and to create multi-domain ASR for recognition of highly varied spoken content for SCR applications. The following two research questions (RQs) are proposed:

- RQ1: Can incorporation of visual features be used for improvement of ASR transcription accuracy?
- RQ2: Can semi-supervised training and acoustic model adaptation improve ASR accuracy for content from diverse uncontrolled topical domains?

The availability of high quality pre-trained models from computer vision (Section 2.2.1) creates the possibility of fusing high level visual features with ASR systems. This thesis investigates application of multimodal techniques to the hybrid HMM-DNN ASR system. As mentioned in Section 2.3, semi-supervised training have been actively investigated, but not in the context of creating a multi-domain ASR system. This thesis investigates the effectiveness of these approaches to developing a multi-domain ASR system for highly varied user-generated data.

Chapter 3

Information Retrieval Models

The goal of Information Retrieval (IR) systems is to satisfy user information needs whereby matching user queries to relevant documents. The key challenge here is how to score documents by their likelihood of being relevant to the information need. Early IR systems employed Boolean search where a query was a logical combination of terms and a set of returned documents were the exact match with the query (Sanderson and Croft, 2012). The concept of ranking documents according to their potential relevance emerged when the vector space model was introduced by Salton et al. (1975). The vector space model represents documents and queries as vectors which is the concept still used these days. Following the emergence of the concept of document ranking, the probabilistic IR model called BM25 was introduced (Robertson et al., 1995). The BM25 model is still commonly used to produce an initial ranked list of documents which more effective neural ranking models take as input Guo et al. (2020); Lin et al. (2020).

As with the field of ASR (Chapter 2), recent IR research in the deep learning paradigm has focused on the creation of effective ranking models using neural networks. Thanks to the introduction of the large scale IR dataset for neural model training (Bajaj et al., 2016), ranking models using neural networks, so called neural ranking models have been demonstrated to have more effective search output than the BM25 model (Guo et al., 2020). Neural ranking models conduct retrieval in two stages to avoid increased latency caused by a neural network. The first stage is to apply a sparse word based matching model such as the BM25 model to obtain an initial ranked list of documents. The second stage is to re-rank top k ranked documents in the initial list using a neural ranking model.

The latest IR research demonstrates transformer-based IR is a highly effective alternative to traditional models such as BM25 and even to neural ranking models (Lin et al., 2020). The transformer architecture is a special architecture of neural networks originally proposed for machine translation (Vaswani et al., 2017). In this paradigm, a large scale model using the transformer architecture is pre-trained on language modelling and fine-tuned for the IR task by exploiting distributed knowledge of language accumulated during the pre-training stage. There are two variants of transformer-based IR systems: re-ranking systems (Nogueira and Cho, 2019) and dense retrieval systems (Karpukhin et al., 2020). Briefly, the re-ranking system is similar to neural ranking models which re-score the ranked list of documents returned by a faster matching based model (e.g., BM25). The dense retrieval system vectorises documents into dense vector indexes and vector-based similarities are used to compute query-document relevance scores.

The subsequent sections of this chapter describe three ranking algorithms: BM25, neural ranking models and transformer-based IR systems. Neural ranking models introduced in this chapter are Deep Relevance Matching Model (DRMM) and Position-Aware Convolutional Recurrent Relevance (PACRR) model. Transformer-based systems introduced are the re-ranking system and the dense retrieval (DR) system. The BM25 model provides experimental baseline in our SCR investigation. The impressive results recently obtained using neural ranking models lead us to explore these methods for SCR, and to examine their behaviour when using the speech processing strategies explored in this thesis. Following the introduction of the ranking algorithms, common evaluation metrics used for IR experiments are described. At the end of the chapter two research questions relevant to IR models applied to SCR are proposed.

3.1 The BM25 Probabilistic Model

The BM25 seeks to compute a relevance score of document given a user query and ranks documents in decreasing order of relevance scores (Robertson et al., 1995). The BM25 model was derived from a theoretical background of IR within a probabilistic framework, with a model derived based on some practical approximations (Robertson and Walker, 1994). The BM25 relevance score for each document is calculated by summing the weights of words from the query present in each document. The weights of words are designed as a measure of the importance of the words with respect to the relevance of the document to the user's information needs as described in the query. The BM25 weight for each word is calculated as shown in Equation 3.1 and Equation 3.2.

$$score(d,q) = \sum_{i=1}^{n} IDF(q_i).\frac{f(q_i,d).(k_1+1)}{f(q_i,d)+k_1.(1-b+b.\frac{|d|}{avgdl)})} \tag{3.1}$$

$$IDF(q_i) = log\frac{N-n(q_i)+0.5}{n(q_i)+0.5} \tag{3.2}$$

where f(q_i, d) is the term frequency of a query word q_i in the document d, $|d|$ is the total number of words in the document d (document length), and $avgdl$ is the average document length of the collection of documents, k_1 and b are parameters to weight term frequency and normalise document length variations respectively. $IDF(q_i)$ is an inverse document frequency (IDF) of a query word q_i that shows uniquness of the word in a document collection. $n(q_i)$ is the total number of documents containing term q_i and N is the total number of documents in the collection.

3.2 Neural Ranking Model

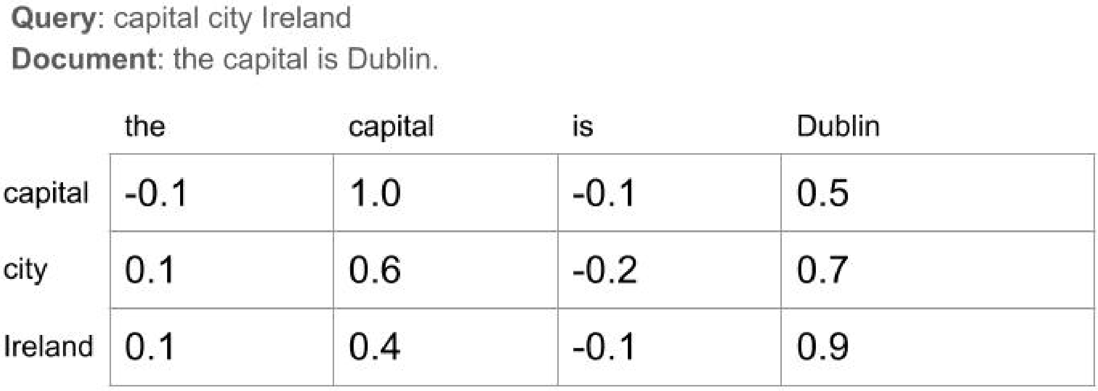
The ranking model is a core algorithm of IR methods. There has been active research on the use of neural networks for the ranking models to improve search effectiveness over the probabilistic models such as the BM25 model (Guo et al., 2020). Ranking models using neural networks are referred to as *neural ranking models*. Typically, training a neural network model requires a large amount of manually created labels. Development of neural ranking models is supported by the availability of the large scale Microsoft MAchine Reading COmprehension (MS MARCO) dataset that contains over 100k training queries constructed from Bing search (Bajaj et al., 2016). The TREC Deep Learning Track organised since 2019 has demonstrated search effectiveness of neural network-based models over the BM25 model (Craswell et al., 2021).

The neural ranking model takes a query and document pair, and outputs a relevance score of the input pair. While the neural ranking model has been found to be more effective than the BM25 model, its inference to compute relevance scores is slower than the BM25 model. For this reason, the neural ranking model is typically applied to re-ranking of a limited number of ranked documents returned by the BM25 model or another word count based model.

The following sub-sections introduce two neural ranking models: a deep relevance matching model (DRMM) (Guo et al., 2016) and Position Aware Convolutional Recurrent Relevance (PACRR) matching (Hui et al., 2017). We chose DRMM and PACRR for our investigation, since these models are relatively new models in the neural IR paradigm before transformer ranking models, and DRMM and PACRR are also simple to implement. These two neural ranking models belong to the group of interaction-based models where the input to the model is the interaction of a query-document pair, unlike representation-based models including Deep Structured Semantic Models (Huang et al., 2013a), where representations of a query and a document are computed independently and the final relevance score is computed by analysing the two representations.

3.2.1 Similarity Matrix

As mentioned in Section 3.2, the DRMM and PACRR models belong to the interaction-based models. An important concept to discuss with respect to interaction-based



Query: capital city Ireland
Document: the capital is Dublin.

	the	capital	is	Dublin
capital	-0.1	1.0	-0.1	0.5
city	0.1	0.6	-0.2	0.7
Ireland	0.1	0.4	-0.1	0.9

Figure 3.1: An example similarity matrix constructed from the query “capital city Ireland” and the document “the capital is Dublin”.

models is the *similarity matrix* that represents a query-document interaction. Suppose a query consists of Q terms with $q = w^q_1 w^q_2 ... w^q_Q$ and a document of its length D consists of $d = w^d_1 w^d_2 ... w^d_D$, query-document interactions are represented as a matrix of similarity of each query term against each document term. Both query terms and document terms are typically represented as a fixed dimensional word vector (Pennington et al., 2014), and a similarity matrix is created by computing the cosine similarity of each query term against document term. The similarity matrix therefore has the size $S \in [-1, 1]^{|Q| \times |D|}$.

Figure 3.1 illustrates an example similarity matrix created from the query “capital city Ireland” and the document “the capital is Dublin”. The matrix size is 3×4 because the query length is 3 and the document length is 4. The numbers in each cell indicate a similarity score of a query term w_i against a document term w_j. For example, the “capital” is the term present in both the query and the document and its the cosine similarity of the two becomes 1.0, while the cosine similarity of semantically different terms “Ireland” and “is” goes to a negative value.

3.2.2 Deep Relevance Matching Model

Figure 3.2 illustrates how the Deep Relevance Matching Model (DRMM) computes a relevance score of a given query and document. There are three stages for relevance score computation. The first stage is creation of a similarity matrix, described in

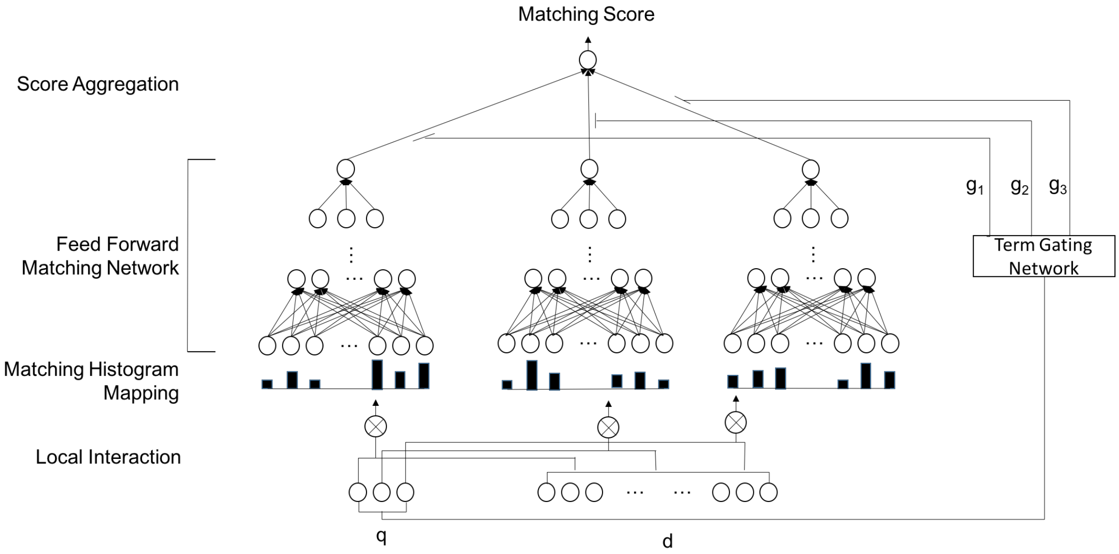
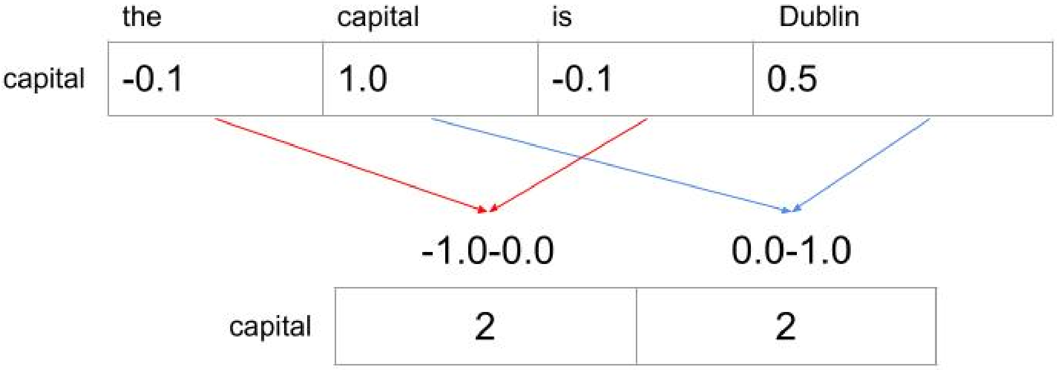


Figure 3.2: Diagram showing how DRMM computes a query-document relevance score; adopted from Guo et al. (2016).

Section 3.2.1, and transformation of the matrix into a matching histogram. The second stage is obtaining hidden representations of matching histograms using a DNN and computing query term weights using a single layer neural network and the attention mechanism (Bahdanau et al., 2015). Finally, the hidden representations and the term weights are aggregated to produce a relevance score. The rest of this section outlines each stage of relevance score computation using DRMM.

Matching histogram

The matching histogram is an input representation of an DNN for DRMM. This operation of transforming a similarity matrix into a matching histogram allows a DNN to take as input vectors of the same size for variable length documents. The matching histogram is created by transforming each row of the similarity matrix into a histogram with b bins according to values of the cosine similarity. Figure 3.3 illustrates an example of transforming one row of a similarity matrix into a matching histogram with 2 bins. Since the value of the cosine similarity ranges in $[-1, 1]$, the values equal to and less than 0 are counted in the first bin and other values are in the other bin. Therefore, the resulting matching histogram of this example contains 2 in the first bin and also 2 in the other bin. Guo et al. (2016) found that applying



	the	capital	is	Dublin
capital	-0.1	1.0	-0.1	0.5

	-1.0-0.0	0.0-1.0
capital	2	2

Figure 3.3: An example of transforming one row of a similarity matrix into a matching histogram.

a logarithm to each frequency bin of the matching histogram improved retrieval results.

Gating weight

The gating weight is used to measure "importance" of each query term. Suppose a query is "capital city Ireland", a proper noun "Ireland" is more important than "city" in identifying relevant documents. The input of the term gating network which consists of a single layer neural network and the attention mechanism (Bahdanau et al., 2015) is the gating vector consisting of word embedding vectors of query terms concatenated with IDF values (Equation 3.2) of the query terms (McDonald et al., 2018). The gating weight g_i, a weight for ith query term, can be computed by applying the attention mechanism to the gating vector $\mathbf{v}$ as shown in Equation 3.3.

$$g_i = \frac{exp(\mathbf{w}_a \mathbf{v}_i)}{\sum_{i=1}^{Q} exp(\mathbf{w}_a \mathbf{v}_i)} \tag{3.3}$$

where $\mathbf{v_i}$ is a gating vector of ith query term consisting of its word embedding (e.g., word embedding vector of "Ireland") and IDF value, $\mathbf{w_a}$ is a weight vector for the attention layer and $\mathbf{b}$ is a bias term for the attention layer.

Relevance score computation

As shown in Figure 3.2, the final relevance score is computed by feeding the matching histograms to a DNN model and aggregating the output of the DNN with the gating weights. Suppose the neural network consists of Z linear layers, the relevance score o is computed as shown in Equation 3.4 to Equation 3.6.

$$\mathbf{m}_i^{(0)} = \mathbf{m}_i \tag{3.4}$$

$$\mathbf{m}_i^{(z)} = tanh(\mathbf{W}^{(z)}\mathbf{m}_i^{(z-1)} + \mathbf{b}^{(z)}) \tag{3.5}$$

$$o = \sum_{i=1}^{Q} g_i \mathbf{m}_i^{(Z)} \tag{3.6}$$

where $\mathbf{m}_i^{(z)}$ is a hidden representation of the ith row of a matching histogram after the zth layer, $\mathbf{W}^{(z)}$ the weight matrix of the zth layer, $\mathbf{b}^{(z)}$ the bias term of the zth layer and g_i the gating weight for a query term j.

3.2.3 Position Aware Convolutional Recurrent Relevance Matching

The second neural ranking model to be introduced in this section is the Position Aware Convolutional Recurrent Relevance Matching (PACRR) model. Figure 3.4 illustrates computation of a relevance score using PACRR. There are three stages to compute a relevance score using PACRR: convolution, pooling and computation of relevance score. While DRMM transforms an input similarity matrix into matching histograms (Section 3.2.2), the PACRR model directly applies convolutional operations to a similarity matrix to obtain hidden representations of the query-document interaction. The advantage of the PACRR model over DRMM is that PACRR exploits word orders preserved in the similarity matrix when creating the hidden representations. On the other hand, unlike DRMM, PACRR needs to handle variable length documents as the hyper-parameter l_d and documents longer than l_d are cut-off to contain l_d terms or padded to l_d terms if documents are shorter. The similarity matrix therefore has the size of $S \in [-1, 1]^{|Q| \times |l_d|}$. The rest of this sub-section describes the three stages of PACRR to compute a final relevance score.

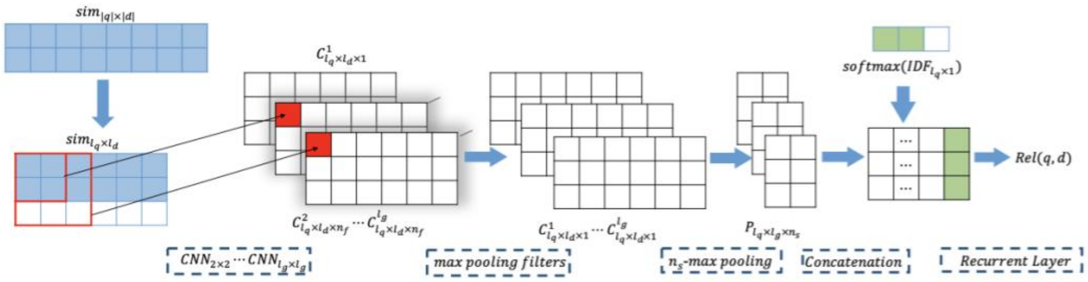

$sim_{|q|\times|d|}$
$sim_{l_q \times l_d}$
$C^1_{l_q \times l_d \times 1}$
$C^2_{l_q \times l_d \times n_f} \cdots C^{l_g}_{l_q \times l_d \times n_f}$
$CNN_{2\times2} \cdots CNN_{l_g \times l_g}$
$C^1_{l_q \times l_d \times 1} \cdots C^{l_g}_{l_q \times l_d \times 1}$
max pooling filters
$P_{l_q \times l_g \times n_s}$
n_s-max pooling
Concatenation
$softmax(IDF_{l_q \times 1})$
...
...
...
$Rel(q,d)$
Recurrent Layer

Figure 3.4: Diagram of how the PACRR model computes a query-document relevance score; adopted from Hui et al. (2017).

Convolution

The core operations to produce hidden representations from a similarity matrix are convolution and pooling (Hui et al., 2017). There are two hyper-parameters to adjust convolution operations for PACRR: the number of convolutional layers l_g and the size of output values for each convolutional operation l_f. When the l_g parameter is set to 3, two convolutional layers with kernel sizes 2×2 and 3×3 are applied to the similarity matrix (kernel size 1×1 passes all of the values in a similarity matrix). The $n \times n$ kernel size extracts N-gram similarity of query and document terms from the similarity matrix. For example, kernel size of 2×2 is the bi-gram sub-matrix of query and document terms. This is an important concept of PACRR to exploit word orders present in the similarity matrix. The convolutional layer proceeds in operation with the stride size (1, 1), meaning that the filter moves 1 step at a time. Further, padding is applied to the similarity matrices, so that the output of the convolution operations of PACRR retains the size of the original similarity matrix.

Formally, suppose a convolutional layer has kernel size k, the convolutional operation can be expressed as shown in Equation 3.7.

$$C^k_{l_f} = Conv^k(S) \tag{3.7}$$

where S is an input similarity matrix, $Conv^k$ is a convolutional layer with kernel size k and $C^k_{l_f}$ is an output feature tensor of the convlutional layer. $C^k_{l_f}$ is a 3D tensor of size $Q \times l_d \times l_f$ (Q is the query length and l_f the size of output for each convolutional operation). As mentioned above, the convolutional layer with kernel size 1 will output the input similarity matrix as it is and when $k = 1$, $C^1 = S$. The convolutional operations therefore lead to $C^1, C^2_{l_f}, ..., C^{l_g}_{l_f}$.

Pooling

The goal of pooling is to extract salient features from the output of the convolutional operations (Hui et al., 2017). The pooling of PACRR consists of two max pooling layers. The max pooling retains the most salient values over the filter dimension for $C^2_{l_f}, ..., C^{l_g}_{l_f}$. This operation is skipped for C^1, since this tensor does not contain the filter dimension. This produces l_g 3D tensors of size $Q \times l_d \times 1$. The second pooling is controlled by the new hyper-parameter l_s. The l_s parameter determines the number of values to retain after second max pooling. This pooling is applied to the dimension of each l_g-gram tensor, followed by concatenation of these tensors to form a matrix of the size $Q \times (l_g \times l_s)$. This is an input matrix of the neural network model that computes the final relevance score.

Relevance score computation

The final relevance score is produced by a few layers of the neural network model given output from max pooling. Similar to DRMM (Section 3.2.2), the gating weights are also used for the PACRR model. Although the original paper describing PACRR suggests use of an RNN (Section 2.1.3), follow-up work by McDonald et al. (2018) found that replacing the RNN with a simple feed-forward network was effective. Further, both Hui et al. (2017) and McDonald et al. (2018) concatenate the gating weights with the output of the pooling operations, Moriya and Jones (2021a) found that it was empirically better to apply a feed-forward network to the pooling output and aggregate output of the network with the gating weights. This is essentially how a relevance score is computed in DRMM using the weights from the gating vector shown in Equations 3.4-3.6, except that the non-linear function used is ReLU instead of the hyperbolic tangent, as suggested by Hui et al. (2017).

3.2.4 Training neural ranking models

To train neural ranking models, a triplet of a query, a document relevant to the query and a document irrelevant to the query (q, d^+, d^-) is taken as input to the model (Guo et al., 2016; Hui et al., 2017). The model is trained to produce a higher relevance score for relevant query-document pairs q and d^+ using a pair-wise ranking loss. In other words, the model is not directly optimised to increase the retrieval metrics. This can be expressed in Equation 3.8

$$L(q, d^+, d^-; \theta) = max(0, 1 - o^+ + o^-) \tag{3.8}$$

where θ is the model parameters, o^+ is output of the model from a true query-document pair and o^- is output of the model from a negative query document pair. To monitor training progress of the neural ranking model with regard to the chosen IR metrics, a validation set can be used to compute the metrics after every interval of model training.

3.3 Transformer-based Model

Recent research on the IR field suggests that transformer-based ranking models are highly effective alternatives to traditional probabilistic models and neural ranking models (Lin et al., 2020). The core concept of transformer-based ranking models is to fine-tune a pre-trained large language model (LLM) using the transformer architecture to an IR task. The first LLM proposed and commonly used for fine-tuning is the Bidirectional Encoder Representation from Transformer (BERT) (Devlin et al., 2019). As mentioned in introduction of this chapter, two transformer-based IR systems are commonly used: re-ranking and dense retrieval (DR). This section first presents an overview on the BERT model as a prerequisite of introducing transformer-based ranking models and then introduces transformer-based re-ranking and DR models.

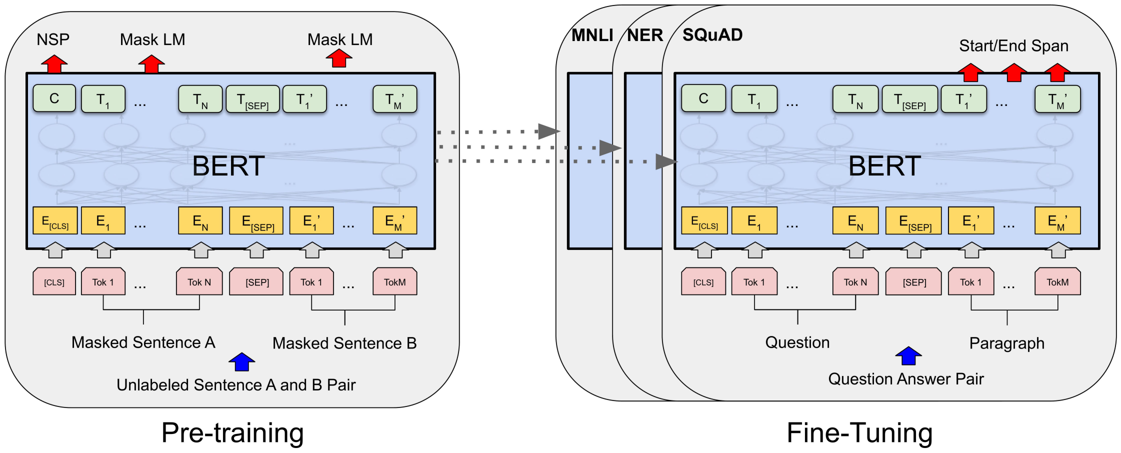


Figure 3.5: Diagram showing BERT pre-training and fine-tuning; adopted from (Devlin et al., 2019)

3.3.1 BERT overview

This sub-section presents an overview on the BERT model that is a basis of transformer-based ranking models (Lin et al., 2020). Devlin et al. (2019) demonstrate that BERT pre-trained on a large amount of unlabelled texts can be fine-tuned to achieve state-of-the-art results for many NLP tasks including question answering and named entity recognition. This sub-section overviews BERT's architecture, pre-training methods and input structure that is relevant to transformer-based search models.

BERT architecture

The BERT model consists of several transformer blocks that incorporate feed-forward layers and multi-head self-attention layers (Devlin et al., 2019). The transformer architecture was originally proposed for machine translation and demonstrates superior performance over recurrent neural networks and traditional attention mechanism using a single layer network to compute weights of input (Vaswani et al., 2017). Briefly, self-attention is a technique to compute a representation from an input sequence by relating different positions of the same input sequence. Multi-head attention uses different representation sub-spaces to compute the final representation rather than a single output of attention.

BERT pre-training and fine-tuning

The important concepts of BERT relating to the transformer-based ranking model are pre-training and fine-tuning (Lin et al., 2020). Pre-training enables a transformer-based model to learn a distributed knowledge of language and fine-tuning updates model parameters to a target task. Figure 3.5 shows a pre-training and fine-tuning scheme for BERT. BERT is pre-trained on two different tasks: masked word prediction and next sentence prediction. Masked word prediction is to predict words hidden in an input sequence given the contexts of the rest of an input sequence. Next sentence prediction is a task to determine if two sentences are next each other in a document.

BERT fine-tuning is carried out by adding a single dense feed-forward layer to the transformer blocks whose weights are adjusted by pre-training. BERT employs two special tokens for fine-tuning of downstream tasks including question answering, named entity recognition and retrieval. The first token is the $[CLS]$ token that precedes all of the input tokens in a sequence. The $[CLS]$ token is an important concept as the representation corresponding to the $[CLS]$ token is the aggregated representation of an input sequence. The second token is the $[SEP]$ token. The $[SEP]$ token is used to separate two word sequences. For example, a sequence of words forming a question and another sequence of words forming a candidate answer can be separated by this $[SEP]$ token. In the case of next sentence prediction for example, an example input sequence can be $[CLS]token_1, ..., token_N[SEP]token_1, ..., token_M$. The dense feed-forward layer added to the BERT model is a binary classification layer (i.e., two sentences next to each other or not) and the final probability of is obtained by feeding the final representation of the $[CLS]$ token to the classification layer. Due to this input format, the length of BERT input is generally constrained to the certain number of tokens. An input sequence less than the length limit is padded to the specified length and tokens exceeding the length limit are cut-off from a sequence longer than the limit.

3.3.2 Transformer ranking models

The transformer-based ranking models are shown to be effective for IR tasks and often outperform traditional sparse matching models (e.g., BM25) and neural ranking models (Lin et al., 2020). The transformer-based ranking models can be classified into two types: re-ranking systems and dense retrieval (DR) systems. Re-ranking systems are similar to neural ranking models that re-score a ranked list of top k documents returned by traditional probabilistic model. Ranking the whole document collection using BERT leads to long run-time, and the re-ranking method improves computational efficiency. The DR system on the other hand employs a transformer model to vectorise a document collection in a dense vector index. This system requires to encoding only a query into a vector representation using a transformer-based model and a fast vector similarity algorithm can then create a ranked list of documents at search time. This section overviews transformer-based re-ranking and DR models.

Transformer re-ranking model

The transformer-based re-ranking model was proposed by Nogueira and Cho (2019). They employ a pre-trained BERT model and fine-tune it for the IR task. The BERT model for re-ranking is referred to as "MonoBERT". The MonoBERT model takes as input a query and a document and directly returns a relevance score of the query-document pair. Figure 3.6 shows an overview process of the BERT re-ranking system. The first search model using a faster algorithm (e.g., BM25) is used to produce an initial ranked list of documents and MonoBERT is used to re-rank the initial ranked list due to slow speed of inference. To compute the relevance score of a query-document pair, an input sequence separates a query from a document using the $[SEP]$ token introduced in Section 3.3.1. Suppose a query consists of N tokens and a document M tokens, an input sequence of the MonoBERT is $[CLS]q_1, ..., q_N[SEP]d_1, ..., d_M$ where q_i is a query token at ith position and d_j is a document token at jth position.

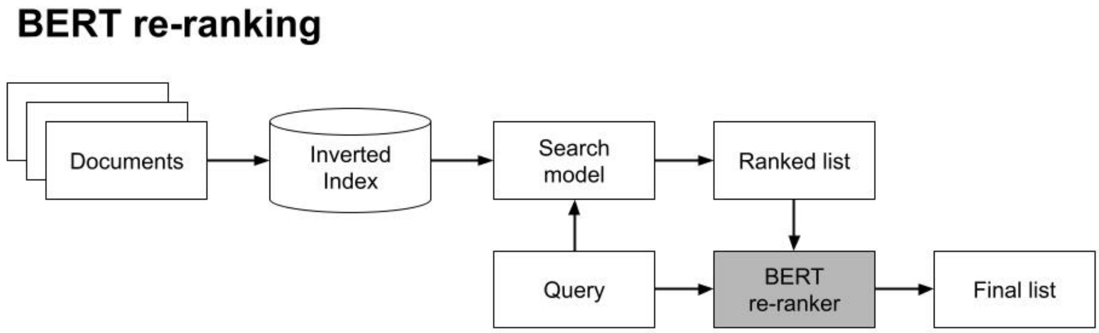


Figure 3.6: An overview of the re-ranking system using BERT.

Transformer DR model

The second type of transformer-based ranking model is a dense retrieval (DR) model (Karpukhin et al., 2020). The DR model is used to encode queries and documents separately, unlike the re-ranking model. Figure 3.7 illustrates the DR system using a BERT model. In the DR system, the BERT model is used to encode documents to create a vectorised index. Queries are encoded by the same model separately to create query vectors. An efficient vector similarity model is used as a ranking model to compare a query vector against the vectorised documents and produces the final ranked list of documents.

Suppose a document consists of M tokens, $[CLS]d_1, ..., d_M$ is an input of the DR model and a vector output corresponding to the $[CLS]$ token is stored as a dense vector representation of the document. At search time, a query input $[CLS]q_1, ..., q_N$ is encoded using the transformer model and a similarity measure is used to compute relevance score of the query representation against document representations in an index. A similarity score of a query-document pair is often computed using approximate nearest neighbour (ANN) search (Karpukhin et al., 2020; Xiong et al., 2021). While the DR approach is 100 times faster than the re-ranking approach according to Xiong et al. (2021), the literature shows these re-ranking systems generally produce better IR metric results (Lin et al., 2020).

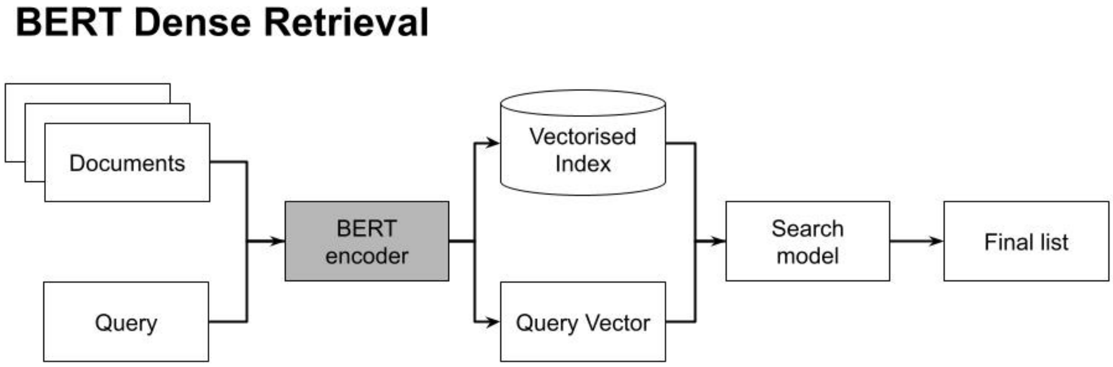


Figure 3.7: An overview of the dense retrieval system using BERT.

3.3.3 Training transformer ranking models

Training of the transformer-based ranking models differs between the re-ranking model and the DR model. The re-ranking model employs a cross entropy loss for model training. Specifically, the re-ranking model is trained for a binary classification task to determine whether a given query-document pair is relevant or not (Nogueira and Cho, 2019). Suppose q_i is ith query and d_i is ith document in train data, the cross entropy loss is computed as in Equation 3.10

$$s_i = g(q_i, d_i; \theta) \tag{3.9}$$

$$L = -\sum_{i=1}^{N} r_i \log(s_i) + (1 - r_i) \log(1 - s_i) \tag{3.10}$$

where $g()$ is a function to compute a relevance score given a model parameter, query and document, r_i is a relevance score of ith query-document pair. The binary cross entropy encourages a BERT re-ranking model to produce a relevance score of 1 for a pair of query and relevant document, while a relevance score 0 for a pair of query and irrelevant document. The difference of binary cross entropy from pair-wise ranking loss (Section 3.2.4) is that one example of pair-wise ranking loss is a triplet of query, relevant document and irrelevant document, while one example of binary cross entropy is a query-document pair and its relevance is fed to the loss as a label.

The transformer DR model is, on the other hand, trained using negative log likelihood to maximise a similarity score of vector representations of a query-document pair (Karpukhin et al., 2020; Xiong et al., 2021). Suppose a query is q, a relevant document is d and a model parameter is θ, the similarity score of the query-document pair can be computed as in Equation 3.11.

$$s(q, d) = sim(g(q; \theta), g(d; \theta)) \tag{3.11}$$

where $g()$ is a function to produce a vector representation of input using the transformer model. Suppose a d^+ is a document relevant to a query and d^- is a irrelevant document, a negative log likelihood of this triplet is computed as in Equation 3.12.

$$L = -\log \frac{\exp(s(q, d^+))}{\exp(s(q, d^+)) + \exp(s(q, d^-))} \tag{3.12}$$

Xiong et al. (2021) provides theoretical grounds of improving selection of negative examples for training a DR model. Their training method called Approximate nearest neighbour Negative Contrastive Estimation (ANCE) updates irrelevant documents for each query-document pair at every checkpoint of a model being trained. Unfortunately, updating negative examples for training of the re-ranking model is not feasible because this would require to compute a relevance score for each query against every document in the collection.

3.4 IR evaluation

In IR experiments, documents relevant to queries are prepared beforehand, so that different ranking models can be compared in the same settings. This section reviews mean reciprocal rank (MRR), mean average precision (MAP) and normalised discounted cumulative gain (nDCG) which are commonly used metrics in evaluation of IR methods.

3.4.1 Mean reciprocal rank (MRR)

The MRR metric is conventionally used for known-item search tasks, where each query has a single relevant document. This evaluation metric was employed for the TREC-6 Spoken Document Retrieval Track (Garofolo et al., 1997) and for the MediaEval Search and Hyperlinking 2012 (Eskevich et al., 2012). The MRR metric is defined as in Equation 3.13:

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \tag{3.13}$$

where N is the number of user queries and $rank_i$ is the rank of the document relevant for the ith query.

3.4.2 Mean average precision (MAP)

The MAP metric is commonly used for ad-hoc retrieval tasks, where each query has multiple relevant documents. This evaluation metric was used for the TREC-7-9 Spoken Document Retrieval Tracks (Garofolo et al., 2000) and for the MediaEval Search and Hyperlinking 2013 and 2014 (Eskevich et al., 2013a, 2014). The MAP metric is defined as in Equation 3.4.2.

$$AP = \frac{1}{M}\sum_{r=1}^{M}\frac{1}{rank_r} \tag{3.14}$$

$$MAP = \frac{1}{N}\sum_{i=1}^{N}AP_i \tag{3.15}$$

where M is the number of relevant documents of ith query, $rank_r$ is the rank of the rth relevant document and N is the number of queries.

3.4.3 normalised discounted cumulative gain (nDCG)

The nDCG metric is used for experiments where each relevant document has a different weight. For instance, a single query can have two relevant documents and one document can be more relevant to the query than the other document. The nDCG metrics was used for the TREC Podcasts Tracks (Jones et al., 2020, 2021). nDCG is the normalised version of the discounted cumulative gain (DCG) as shown in Equations 3.16-3.18.

$$DCG = \sum_{i=1}^{P} \frac{rel_i}{\log(i+1)} \tag{3.16}$$

$$IDCG = \sum_{i=1}^{|REL|} \frac{rel_i}{log(i+1)} \tag{3.17}$$

$$nDCG = \frac{DCG}{IDCG} \tag{3.18}$$

where rel_i is the weight of ith document, DCG is the discounted cumulative gain of retrieved documents with weights, $IDCG$ is the ideal version of the DCG score.

3.5 Research questions

Although neural ranking models have been applied to retrieval of textual documents, little has been investigated in terms of the effectiveness of neural ranking models for SCR. The TREC Podcasts Tracks in 2020 and 2021 have seen the use of transformer-based ranking models (Jones et al., 2020, 2021). Although the transformer-based ranking models outperformed the traditional probabilistic models in the tasks, manual transcripts of podcasts were not available in the tasks and it was not clear whether those models could overcome ASR transcription errors with their efficiency to model word sequences given contexts. The research questions proposed in this thesis relevant to retrieval models are as follows.

Despite the use of transformer-based models in TREC Podcasts tracks (Jones et al., 2020, 2021), it is of interest to investigate the effectiveness of neural ranking models and transformer-based models on both ASR transcripts and manual transcripts. Another research question proposed examines whether extending neural ranking models using N-best hypotheses can increase search effectiveness.

- RQ3: How effective are neural ranking and transformer ranking models for SCR?

- RQ4: Can current neural models be extended for improved SCR effectiveness, e.g. using N-best ASR hypotheses to increase search effectiveness

Chapter 4

Overview of Existing Work on Spoken Content Retrieval

Starting from the TREC Spoken Document Retrieval Track (Garofolo et al., 2000), a number of benchmarking shared tasks have been organised examining Spoken Content Retrieval (SCR) tasks. The benchmarking tasks have generally been of increasing complexity, and have promoted the proposal of new approaches to SCR which seek to address the unique challenges of SCR.

The size of the corpora used for the benchmarking tasks has grown over the time, with the latest TREC Podcasts Tracks providing participants with over 100,000 podcast episodes corresponding to over 50,000 hours of audio (Jones et al., 2020). This chapter overviews the most relevant of these previous SCR tasks with the goal of providing readers with sufficient background of previous work in SCR and the challenges identified to motivate the research reported in this thesis. This chapter focuses on the following SCR benchmark tasks:

- TREC 6-9 Spoken Document Retrieval Track
- MediaEval 2012-2014 Search and Hyperlinking Task
- TREC 2020-2021 Podcasts Track
- CLEF-SR 2005-2007

- NTCIR SpokenDoc 2011 and 2013

4.1 TREC Spoken Document Retrieval Track

The first benchmark tasks focusing on SCR was carried out in the TREC 6-9 Spoken Document Retrieval (SDR) Tracks from 1997 to 2000 (Garofolo et al., 2000). This campaign began with the motivation of improving accessibility of ever growing spoken audio archives and developing a shared community of the effectiveness of the state-of-the-art methods. The corpora used for the campaign consisted of broadcast radio and TV news. Most of the speech is largely without background noise, has a formal speaking style, is in the forcus of monologues and is thus relatively easier for ASR systems to recognise than more naturally occurring speech.

TREC-6 SDR

The first SDR Track in TREC used a corpus of around 50 hours with 1,451 episodes (Garofolo et al., 1997). The organiser provided participants with manually created transcripts and ASR transcripts generated by the IBM system (Dharanipragada et al., 1998). The IBM ASR transcripts were created using a GMM-based system and produced the WER 50%. The participants also had access to raw speech data from the corpus, so that they could develop either a complete pipeline of the SDR system combining ASR and IR (*Full SDR*) or an independent IR system only applied to the provided transcripts (*Quasi-SDR*). There were 47 test queries available. The relevant documents corresponding to half of the queries were designed to be "easy-to-recognise" (clean speech from a native speaker) and the other half to be "difficult-to-recognise" (telephone channels, non-native speakers or speech with noise or music in the background).

The retrieval task conducted was a *known-item search* task where each query is associated with a single relevant document. This is a simpler task than *ad-hoc search* where a retrieval system needs to return all of an unknown number of relevant documents for given query. The organisers expected that results of ad-hoc search for spoken documents would be too poor because it was the very first large scale study of spoken document retrieval (Garofolo et al., 2000). The evaluation metrics used to measure was Percent Retrieved at Rank 1 and Mean Reciprocal Rank (MRR) (Section 3.4.1). Percent Retrieved at Rank 1 is the proportion of relevant documents being ranked as the first document in the ranking. Percent Retrieved at Rank 1 is a more strict metric than MRR in that a retrieval system is not assigned a point credit, even if a relevant document is ranked as the second document in the retrieved ranked list. To the organiser's surprise, the results of the benchmark were successful with the best system achieving Percent Retrieved at Rank 1 78.7% using the manually created transcripts and 63.8% using the automatic transcripts from IBM (Garofolo et al., 2000). Retrieval scores improved further when using the MRR metrics with the best system achieving higher than 80.0% rate using the manually created transcripts.

TREC-7 SDR

The TREC-7 SDR Track was an extension of the TREC-6 SDR Track (Garofolo et al., 2000). The size of the corpus was nearly doubled and the number of episodes was increased from 1,451 to 2,866. The retrieval task was changed from known-item search to ad-hoc search where all of the potentially relevant documents needed to be retrieved unlike know-item search. These changes introduced additional difficulties into the SDR task. There were 23 test queries created for the ad-hoc search task. There were a wider variety of transcripts used for the TREC-7 SDR Track. The organisers provided participants with three different types of transcripts: (i) closed captions as reference transcripts, (ii) a transcript created with an optimised ASR system created using the CMU Sphinx tool with a WER of 33.8% (Pallett et al., 1999), and (iii) one created with an unoptimised ASR from the same group with a WER of 46.6%. In addition to the transcripts made available to participants, the organisers allowed participants to use three other types of transcripts: (iv) transcripts from participant's own ASR system, (v) transcripts from participant's own second ASR system and (vi) transcripts from other participants' ASR systems. The best ASR transcripts provided from among the participants were from the University of Cambridge with a WER of 24.1% (Johnson et al., 1998).

For this ad-hoc search task, the evaluation metric used was mean average precision (MAP) (Section 3.4.2). In this study, the best retrieval system achieved a MAP of 56.7% using the closed caption transcripts and a MAP of around 50% with the baseline transcripts with WER of 50.6%. The MAP score of the best retrieval system dropped to 41.9% using transcripts of the unoptimised version of CMU Sphinx. From analyses of the submitted runs, it was found that there was a near-linear relationship between WERs and MAP scores (Garofolo et al., 2000). Furthermore, a mean correlation coefficient between recognition accuracy of named entities and MAP scores was even stronger with 0.91. In other words, it is important for ASR systems to transcribe named entities accurately to improve search effectiveness. This latter finding is particularly relevant to the motivation of our work in seeking to improve ASR for SCR.

TREC-8 and TREC-9 SDR

In the TREC-8 and TREC-9 SDR Tracks, the same dataset was used for two years in a row. There were a few changes between TREC-8 and TREC-9 SDR Tracks, including evaluation of the hand-segmented news episodes vs the whole shows, and addition of short description and keyword to each query in TREC-9. The total amount of audio data used for the tasks was 557 hours with 21,754 news story documents. This size was five times larger than than the document set used in the TREC-7 Track. There were 49 test queries available for TREC-8 SDR and 50 test queries for TREC-9 SDR with the aforementioned descriptions and keywords. Similar to the TREC-7 SDR track, five different types of transcripts were used for retrieval. The first baseline ASR system produced a 27.5% WER and the second baseline ASR system a 26.7% WER on 10 hour subset of the speech data. There was a statistically significant difference between the first and the second ASR systems.

The ASR results of the TREC-8 demonstrated that most of the ASR systems provided by participants produced WERs below 30%. The best ASR WER among the participants was WER of 20.5% from the University of Cambridge HTK system (Johnson et al., 1999a). There was a small difference in the MAP scores between the reference transcripts (closed-captions) and the first baseline transcripts, 56.0% and 55.4%. The organisers this year also observed a near-linear relationship between the WERs and the MAP scores. In the TREC-9 SDR Track, there was a decline in participants and only 3 groups participated in the task. In the TREC-9 task, story boundaries were unknown unlike TREC-8. This contributed a small reduction in a MAP score with the best system producing a MAP score slightly higher than 50.0%

TREC SDR Tracks summary

The TREC SDR Track was the first large scale retrieval benchmarking task for spoken documents. The corpus used for the tasks contained broadcast news which is relatively easy for ASR systems to recognise. The MAP scores of 50-55% obtained in the last two years of the task demonstrated that the retrieval systems could run fairly effective search on spoken documents of this form. It was suggested that retrieval of broadcast news could be a "solved problem" (Garofolo et al., 2000), while later Shou et al. (2003) and Sanderson and Shou (2007) demonstrated that the retrieval systems struggle to deal with documents whose WERs are higher. The interesting findings from the TREC-7 SDR Track were a near-linear relationship between WERs and MAP scores, and even higher correlation between recognition accuracy of named entities and MAP scores.

4.2 MediaEval Search and Hyperlinking

An organisation of SCR for Internet videos and diverse broadcast programmes was carried out in the Search and Hyperlinking Task at MediaEval from 2012 to 2014 (Eskevich et al., 2012, 2013a, 2014). The task consisted of two sub-tasks: search sub-task and hyperlinking sub-task (Eskevich et al., 2013b). The goal of the search sub-task was to find the right jump-in point to begin playback of a video given a user query. This would allow users to avoid going through the whole video to find the right start point for playback of content in which they are interested. The goal of the hyperlinking sub-task was to find video segments potentially similar to the jump-in points found in the search sub-task. This would enrich user experience of going through a large video collection. This section overviews the results of the search sub-tasks that are more relevant to the investigations of this thesis. The first year of the Search and Hyperlinking Task used a corpus of semi-professionally edited videos available on the Internet called Blip10000 (Schmiedeke et al., 2013). The last two years of the tasks used broadcast TV materials from the BBC. This section summarises the key findings from the three year campaign of the Search and Hyperlinking Tasks.

The Search and Hyperlinking 2012

The first organisation of the Search and Hyperlinking task in 2012 used a collection of semi-professionally edited videos called *Blip10000* (Schmiedeke et al., 2013). The corpus was divided into 5,288 videos of dev set (1,135 hours) and 9,550 videos of test set (2,125 hours). The corpus contains various types of content ranging from technology, conference talks, street interviews and game commentaries. Compared to the dataset used for the TREC SDR Tracks, the total amount of content was almost 6 times larger and the content type was much broader than broadcast news programmes. The ASR transcripts for the Blip10000 corpus were provided by *LIMSI/Vocapia* (Lamel and Gauvain, 2008) and by *LIUM* (Rousseau et al., 2011). The ASR system from LIMSI/Vocapia incorporated a language identification system. Video transcription was done only when a language identification score was equal to or greater than 0.8. The ASR system from LIUM was on the other hand developed for recognition of English speech and transcripts were provided only when processing audio files succeeded. The submitted systems for the search task were evaluated using three IR metrics: MRR, mGAP and MASP. The mGAP metrics measures the rank of the relevant segment and the error distance of retrieving the jump-in point. The MASP metrics measures the rank of the relevant segment and the quality of segmentation assuming that segmentation is not applied to speech.

The results and findings of the Search and Hyperlinking task 2012 are summarised in (Eskevich et al., 2013b). The best video segmentation approach to achieve the MRR and mGAP scores was time-based segmentation with 90 seconds with an overlap of 30 seconds. This led to the MRR score 47.0% using the language model and the mGAP score 29.0%. The best MASP score was obtained using segmentation based on sentence boundaries from ASR output. The BM25 model produced a 24.0% MASP score. Although test queries included multimodal features (e.g., presence of face, colour features), visual features had a negative impact on the results of search, leading to lower scores for search and segmentation.

The Search and Hyperlinking 2013 and 2014

The Search and Hyperlinking tasks in 2013 and 2014 used a collection of broadcast programme videos from the BBC. Unlike the TREC SDR Tracks, the programmes were not limited to broadcast news and contained a wide variety of programme types. The collection in 2013 consisted of 1,260 hours of videos. The whole collection was used for both development and evaluation. The collection was extended to 4,021 hours of videos in 2014, with the collection from 2013 used for development and 2,686 hours kept for the test set. In 2013, 50 known-item queries, including multimodal features were created and runs submitted by participants were evaluated by humans evaluators. In 2014, the 50 known-item queries from the 2013 tasks were used for system development and 36 ad-hoc search queries were collected from evaluation. While the submitted systems in the 2013 task were evaluated using the same metrics as the 2012 task, the 2014 task was ad-hoc search and employed the MAP metric. Similar to the task in 2012, the ASR transcripts for the datasets were created by LIMSI/Vocapia (Lamel and Gauvain, 2008) and by LIUM (Rousseau et al., 2011).

The main findings from the Search and Hyperlinking tasks in 2013 and 2014 were that the use of closed captions improved search effectiveness, indicating that ASR transcription errors had a negative impact on search results. In the 2013 task, the language model based system submitted by Eskevich and Jones (2013) produced an MRR score 37.6% using closed captions and the score dropped to 31.7% by using LIUM transcripts. In the 2014 task, a MAP score 63.9% was obtained using a TF-IDF based vector space model over the closed-captions (Racca et al., 2014), while the score dropped to 52.5% over the LIMSI transcripts.

Summary

The MediaEval Search and Hyperlinking tasks were the first organised search task seeking to identify relevant "jump-in" segments in a diverse collection of videos. In the 2012 task, the video corpus called Blip10000 (Schmiedeke et al., 2013) containing a diverse set of user-generated content was used. It was found that the choice of segmentation approaches was important to improve search effectiveness. Although multimodal features could be used for search models, those features were not effective for segmentation of videos nor for search operations. The corpus used for the 2013 and 2014 tasks was provided by BBC containing a broad range of broadcast programmes. The results from the 2013 and 2014 tasks demonstrated that there was still a gap in retrieval effectiveness between transcripts with near manual quality (i.e., closed captions) and ASR generated transcripts.

4.3 TREC Podcasts Track

The TREC Podcasts Tracks were organised in 2020 and 2021 (Jones et al., 2020, 2021). The goal of the Podcasts Tracks was to facilitate user access to the ever growing archives of podcast content. There were two tasks of the Podcasts Tracks: retrieval of podcast segments of given a user search query and summarisation of podcast content. This section overviews the findings of the retrieval task relevant to this thesis. The dataset prepared contained over 100,000 episodes of podcasts corresponding to roughly 50,000 hours of audio. Participants were provided with

ASR transcripts created using Google's Speech-to-Text API and also had access to raw audio files. Podcast segments were defined as audio segment of 120 seconds (2 minutes) duration with an 60 second overlap. Relevant segments were evaluated according to 5 different scores: perfect (4), excellent (3), good (2), fair (1) and bad (0). The evaluation metrics used for the challenge was mean nDCG (Section 3.4.3). In the 2020 task, 8 queries were prepared for system development and 50 queries for evaluation. In the 2021 task, a new set of 50 evaluation queries were prepared. Each query was associated with its query type (e.g., topical, known-item) and query description.

Summary

The main finding of the Podcasts Tracks was that BERT and transformer-based retrieval methods outperformed the traditional probabilistic models including the BM25 model. In the 2020 track, the T5 model applied to re-ranking produced nDCG 67.0% which was 8% higher nDCG score than the second best system using the BM25 model (Galuscakova et al., 2020). In the 2021 track, 5 out of 6 participants submitted BERT-based re-ranking or dense retrieval systems (Jones et al., 2021). This finding is notable since BERT retrieval have previously generally been observed in SCR tasks by using established IR models such as BM25. Evaluation of the systems were conducted in 4 different criteria: topical retrieval, entertaining, subjective and discussion. The topical retrieval was a standard topical relevance of returned segments with the best system producing higher than 50% nDCG using a dense retrieval system (Hofstatter et al., 2021). The entertaining criterion was to find segments amusing to listeners. The subjective criterion was to find segments where a speaker expresses their polar opinion about the query topic. The discussion criterion was to find segments where more than one speaker participating in discussion. While the nDCG higher than 40% was achieved for the subjective and discussion criteria, the entertaining was only nDCG 30%, indicating difficulties of finding segments according to the level of entertainment (Jones et al., 2021). The

TREC Podcasts Tracks also did not provide participants with manual transcripts or equivalent of manual transcripts (e.g., closed captions) of podcasts. Therefore, even though the BERT-based systems were found to be effective on this SCR task, it could not be determined to what extent ASR errors contained in the transcripts negatively affected its performance. The results using neural IR models for this SCR task indicate that further investigation of these methods for SCR tasks should be carried out, and also that methods to improve SCR, e.g., relating to ASR, should include analysis of their effectiveness when using neural IR methods since these are likely to give the BERT overall results.

4.4 Other SCR Benchmarks

Two other significant SCR benchmarks were Cross-Language Speech Retrieval (CL-SR) at Cross-Language Evaluation Forum (CLEF) and SpokenDoc at NII Testbeds and Community for Information access Research (NTCIR) campaigns. We only provide a brief summary of these tasks, since these are less relevant to the main content of this thesis than the previous three benchmark campaigns.

4.4.1 CLEF-SR

The CLEF-SR campaign was organised from 2005 to 2007 (White et al., 2006; Oard et al., 2007; Pecina et al., 2008). The CLEF-SR campaign focused on examining SCR for a collection of oral testimonies. While the campaign begun with search of English data in 2005 (White et al., 2006), Czech data was added to the task in 2006 (Oard et al., 2007). The number of documents of the English task was topically coherent 8,104 segments created from interviews. The best available ASR system at that time produced WER of 25% on the English held-out corpus (Pecina et al., 2008). The Czech document collection consisted of 357 interviews that formed 11,377 overlapping passages to be searched. There is no report on the quality of ASR transcripts for the Czech data.

Generally low search effectiveness for the English and Czech tasks were observed. Search for English data was evaluated with MAP, whereas Czech data was not pre-segmented and mGAP that can evaluate partial relevance of retrieved segments was used (Pecina et al., 2008). The highest MAP of the submitted run for the English task was 28.47% and the highest mGAP of the submitted run for the Czech task was 2.74%. This content related to re-telling of previous experiences often making no reference to individuals, places, events, etc. Search of such content was found to be very challenging, and effective search was generally only possible when transcripts were combined with textual metadata.

4.4.2 SpokenDoc at NTCIR

The SpokenDoc benchmark campaign was organised in 2011 and 2013 (Akiba et al., 2011, 2013). This benchmark task focused on retrieval of lecture data. In the 2011 task, the corpus containing 612 hours of Japanese academic presentations was used (Akiba et al., 2011) and two tasks organised were lecture retrieval and passage retrieval given a topic. In the 2013 task, the same Japanese lecture corpus was used for lecture retrieval, while a smaller corpus of 27 hours was used for passage retrieval (Akiba et al., 2013). The organiser provided word-based transcripts and syllable-based transcripts. Throughout the campaign, syllable-based transcripts were around absolute 10% more accurate than word-based transcripts (Akiba et al., 2011, 2013).

These tasks encountered challenges in varied audio quality and informal structure and language, but not topical diversity. As such, it is possible to build domain specific ASR systems to maximise transcription accuracy. Since in our work, we are interested in addressing the challenges of topically diverse spoken archives, we do not consider these tasks further.

Table 4.1: Summary of the SCR benchmark tasks reviewed in this chapter. "year" is years of tasks organised, "size (h)" duration of an audio data collection, "size (n)" the number of documents, episodes or segments of a collection and "data type" the type of audio collection used for the task.

Task	year	size (h)	size (n)	data type
SDR	1997-2000	557	21,754	broadcast news
CLEF-SR (en)	2005-2007	589	8,104	interviews
CLEF-SR (cz)	2006-2007	-	11,377	interviews
NTCIR	2011, 2013	612	2,702	lecture
MediaEval	2012-2014	1,135/2,125	5,288/9,550	user videos
Podcasts	2020-2021	50,000	100,000	podcasts

4.5 Summary and Conclusions

As reviewed in this chapter, a number of SCR benchmarking tasks have been organised since TREC SDR in 1997 (Garofolo et al., 1997). Table 4.1 provides a summary of the SCR five benchmarking tasks reviewed in this chapter. Over the time, the size of a search collection used for benchmarking tasks grew. The "size (n)" column in Table 4.1 shows the number of documents or segments depending on a task. In TREC SDR, NTCIR and MediaEval, "size (n)" is the number of documents in a collection of each task. The MediaEval row shows the size of both dev and test set. In CLEF-SR, "size (n)" corresponds to the number of topical segments created from interviews. The Podcasts collection contained over 100,000 episodes of podcasts and "size (n)" shows this, while search was actually performed on overlapping segments of 120 seconds created from the episodes.

This chapter reviewed three benchmark SCR tasks relevant to this thesis and two other significant benchmark tasks. The main findings from these tasks are summarised as follows:

- A near-linear relationship between WERs and MAP scores and higher correlation between recognition accuracy of named entities and MAP scores (TREC SDR).
- Importance of segmentation approach to finding relevant "jump-in" points (MediaEval Search and Hyperlinking)

- Negative impact of multimodal features on video segmentation and retrieval (MediaEval Search and Hyperlinking)

- Recently proposed BERT-based search models are more effective for SCR Podcast tasks than traditional probabilistic models (TREC Podcasts).

These findings from the benchmarking tasks are relevant to content of this thesis as follows. The linear relationship between WERs and MAP scores found in the TREC SDR Tracks motivates improvement of ASR using visual features (Chapter 5) and semi-supervised training and adaptation of an acoustic model using content genre (Chapter 6). The visual features in particular are hypothesised to improve recognition of named entities given background situational contexts present in visual features. The findings of MediaEval Search and Hyperlinking showed a negative impact of visual features on segmentation and search. This thesis, on the other hand, investigates the use of visual features for ASR systems (Chapter 5). Finally, the latest large scale SCR benchmarking tasks of TREC Podcasts Tracks demonstrated search effectiveness of neural and transformer-based systems. However, it was unclear if transformer-based systems resolved ASR transcription errors for SCR as the manual transcripts or equivalent of manual transcripts were not available in the Podcasts Tracks. This thesis investigates the gap of search effectiveness from neural and transformer-based systems between manual and ASR transcripts and proposes the use of ASR N-best for neural and transformer-based systems to bridge the gap between manual and ASR transcripts.

Chapter 5

Multimodal Augmentation of ASR Systems

In this chapter, we present a research investigation into the use of multimodal information within ASR. We first describe adaptation of an acoustic model and a neural language model using multimodal features. Three types of multimodal features are investigated: (i) visual context features, (ii) embedded video titles and (iii) human face features. While the first two features are relevant to situational contexts and are used for adaptation of a neural language model, the human face feature carries information about speaker demographics and is used for acoustic model adaptation. This chapter provides the details of datasets, ASR architecture and training methods for our experiments, followed by experimental results and analysis using the multimodal contexts for ASR. We end the chapter with discussion and conclusions of our multimodal methods for ASR. The publications relating to these research investigations are listed in the front matter[1].

[1] My contribution is background research, designing and conducting experiments and analysis of the experimental results.

5.1 Integration of a multimodal signal into ASR

As described in Section 2.2, situational contexts surrounding humans have an impact on human speech processing. The majority of work on ASR considers only the speech signal and ignores the situational contexts. Some existing work has though explored visual features of baseball games (Fleischman and Roy, 2008), scene and object features (Gupta et al., 2017), age, gender and race features and spatio-temporal features (Roy et al., 2014). This thesis explores the use of video titles, visual context and speaker face as a fixed-length dense vector for further understanding of the use of various multimodal features for ASR. Advances in computer vision have made many pre-trained computer vision models (Simonyan and Zisserman, 2015; King, 2009) such as object recognisers and face recognisers available. In this investigation we explore the potential for these models to be used to extract visual features from images and use these within ASR systems. The investigations in this chapter explore three multimodal features: (i) visual context features, (ii) video title features and (iii) human face features. The visual context features and human face features belong to the visual feature category while video title features are created using meta-data of a video collection. The first two features are relevant to situational contexts of speech and used for a neural language model, while the human face feature is expected to carry speaker demographics and is used for adaptation of an acoustic model.

5.1.1 Multimodally augmented neural language model

Section 2.1.3 introduced neural language models using the RNN (RNN language model) for ASR N-best re-scoring and ASR lattice re-scoring. This section introduces an extension of a RNN language model using the visual context feature and the video title feature. Figure 5.1 shows the use of the visual context feature and video title feature for an RNN language model. This architecture has been originally used for image caption generation (Vinyals et al., 2015). A pre-trained object

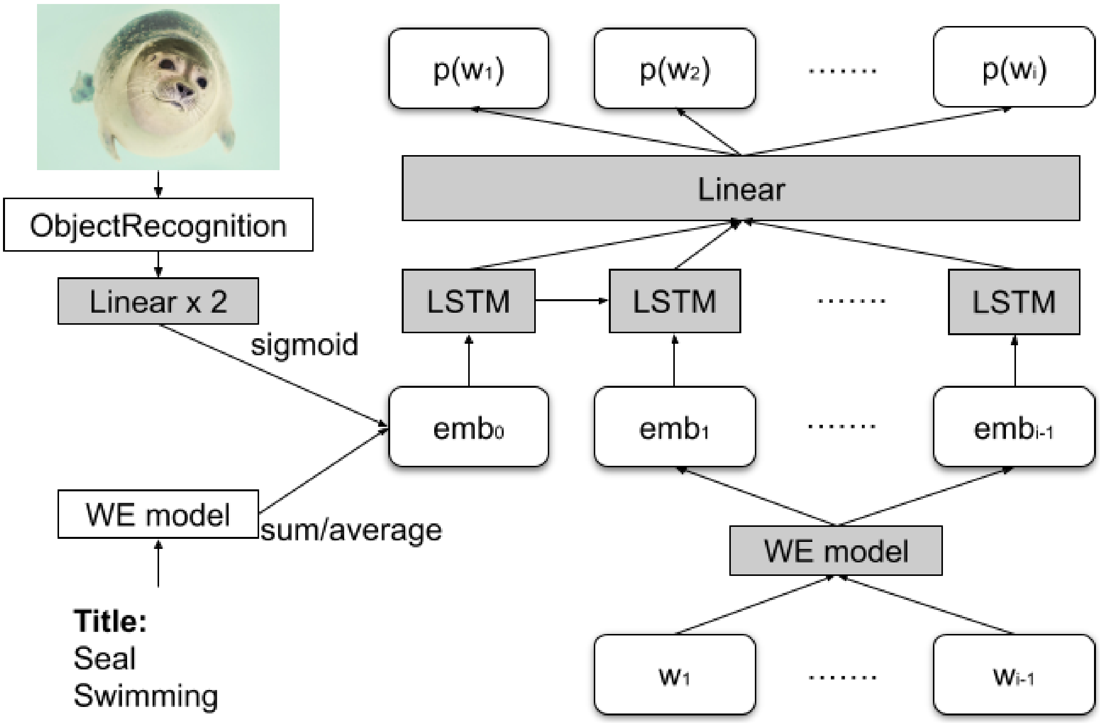


Figure 5.1: Adaptation of a neural language model using the visual context feature and the video title feature. The multimodal feature is taken as the first token of the language model before the first token of a sentence is ready by the model. The grey boxes shown in the figure are trainable parameters.

recognition model is used to extract the visual context feature. The final layer of an object recogniser outputs probabilities of objects likely to appear in an image. The visual feature is extracted from the penultimate layer of the object recognition model. The visual feature goes through two linear layers with the sigmoid function as a non-linear activation function to reduce the size of the visual feature to the size of word embedding vectors. This is input of the neural language model emb_0^i.

Video title features consist of j words $w_1^t, w_2^t ... w_j^t$. Each of the words is transformed into a fixed embedding vector using a word embedding model. There are two approaches available to create a word embedding model that generates a video title feature. The first approach is to use a word embedding model incorporated into an RNN language model (the bottom right "WE model" of the Figure). The second approach is to train a separate word embedding model that is only used for encoding of video titles (the bottom left "WE model" of the Figure). Empirically, we find that it is better to prepare a word embedding model separate from the one incorporated into the neural language model (Results are shown in Section 5.3.1). A single video title embedding is created by taking the average or summation of embedded words to be input to the neural language model emb_0^t. While summation of word embedding vectors has been demonstrated to create a meaningful vector of new concept (Mikolov et al., 2013), averaging word vectors is claimed to be a good sentence representation (Wieting et al., 2016; Arora et al., 2017).

The multimodally augmented RNN language model takes one type of multimodal feature at one time: either the visual context feature or the video title feature. The RNN language model using an LSTM cell can predict words conditioned on the multimodal feature emb_0 (Hochreiter and Schmidhuber, 1997). Suppose a sentence consists of i words $w_1, w_2, ..., w_i$, the multimodal feature is used to predict the first word w_1. This information is propagated to later tokens as a hidden state of an LSTM cell. Output representation from an LSTM cell is fed to one linear layer and softmax is applied to compute probabilities of words.

A multimodally augmented RNN language model can be evaluated using perplexity (Equation 2.7) and used for re-scoring ASR N-best hypotheses (Section 2.1.4). To use a multimodal feature (visual context or video title) for an RNN language model, the multimodal feature is fed to an RNN language model as the first input embedding opposed to the embedding vector of the sentence beginning symbol <sos>. Computation of a score of an ASR hypothesis using a multimodally augmented RNN language model is shown in Equation 5.1.

$$L_{RNN}(h) = -\sum_{i=1}^{I} logP(x_i|x_0...x_{i-1}) \tag{5.1}$$

where h is a hypothesis, I is the number of words in the hypothesis, and x_0 is the visual feature emb_0^i, the video title feature emb_0^t.

A further extension of computation of a score for a single ASR hypothesis is to combine decisions from several language models (both N-gram and RNN) to produce a final score. In this chapter, for example, four different language models will be

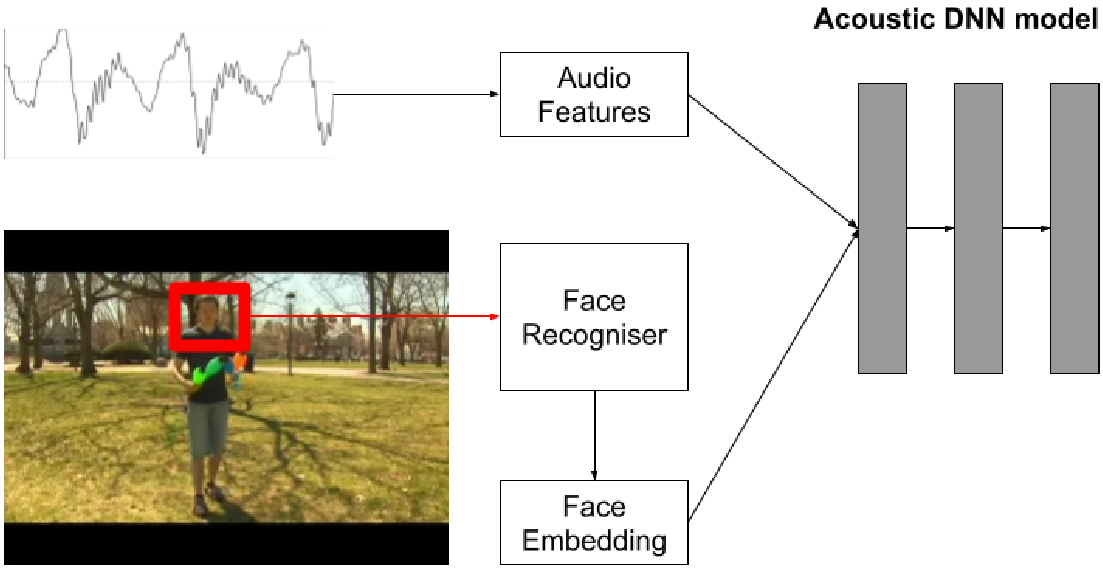


Figure 5.2: Adaptation of a neural acoustic model using the visual speaker face feature. A face recognition model is used to extract speaker face embedding and embedding is concatenated with an acoustic feature vector for input of the neural acoustic model.

trained (N-gram, vanilla RNN language model, visual context RNN language model and video title RNN language model). The final score of an ASR hypothesis using the above four language models can be computed by interpolating scores from the language models and combining it with the acoustic model score as in Equation 5.2.

$$L(h) = AM(h) + \frac{1}{M}\sum_{m=1}^{M} LM(h) \tag{5.2}$$

where $AM(h)$ is the score of the hypothesis of an acoustic model, $LM(h)$ is the score of the hypothesis of a language model, and M is the number of language models to be interpolated. For example, when there are two language models used for score computation, M is set to 2. Since the score is the sum of negative log likelihoods, the lower the score, the better the hypothesis.

5.1.2 Multimodally augmented acoustic model

This section describes the use of speaker face embedding for acoustic model adaptation. The motivation to use speaker face embedding is to enable an acoustic model to implicitly learn speaker demographic information. The use of speaker facial information for ASR has been investigated by Miao et al. (2014). They used the output of a classifier which identifies attributes of a speaker (i.e., “age”, “gender” and “race”) from a visual signal and used probabilities of each attributes from the classification model (e.g., the probability of the person aged 30-40 is 80%). Rather than using predictions from another classification model, this work focuses on dense representation of a speaker face extracted by the face recognition model. This dense representation is hypothesised to be a rich representation and to enable the acoustic model to learn the visual characteristics of a speaker and the acoustic properties of a speaker simultaneously.

An overview of the process to extract a speaker face feature and to use the feature for an acoustic model is illustrated in Figure 5.2. There are two main steps to adapt an acoustic model using a speaker face feature: face embedding extraction using a face recognition model and feeding an acoustic feature vector concatenated with the speaker face feature.

A face recognition model can be pre-trained on a task of classifying faces of popular figures (e.g., actors and politicians). The trained face recognition model learns latent representations of human faces which can distinguish personal differences. This knowledge of the face recognition model can be useful for extraction of the visual face embedding vector. There are two stages to extract speaker face embedding from a video is two stages. The first stage is to identify segments where speaker faces are likely to be present by going through video frames. The second stage is to apply the face recognition model to extract a feature vector from the identified region of a human face. The more details of the algorithm is described in Section 5.2.

The extracted speaker face feature is used as an adaptation vector for an neural acoustic model training. Similar to the i-vector adaptation of the neural acoustic model (Saon et al., 2013), the extracted face feature is concatenated with each acoustic feature vector. The DNN takes as input a concatenation of an acoustic feature vector with a speaker face feature, so that the DNN model can implicitly learn acoustic properties of a speaker and visual attributes (e.g., “gender” and “age”) of a speaker simultaneously.

5.2 Experimental setup

Integration of the visual context feature, video title feature and speaker face feature is investigated in two sets of experiments. While the visual context feature and the video title feature are used to augment the RNN language model, the speaker face feature is used for adaptation of the neural acoustic model. These experiments are carried out to examine contributions of multimodal features with respect to WERs. This section provides the details of datasets used for experiments, descriptions of pre-trained image recognition and face recognition models used as a feature extractor and architecture of the RNN language model and the neural language model.

5.2.1 Datasets

Two different video corpora are used for experiments on multimodal ASR. The first dataset is the How2 dataset (Sanabria et al., 2018) consisting of a collection of user-generated instruction videos. In each video, a speaker demonstrates their expertise including martial art, crafting, cooking, sports and their life experience. The second dataset used is Udacity corpus[2] consisting of lecture videos. The Udacity videos are not simple recordings of lecture videos but multiple people can appear in lectures and conversation skits can be included in these videos. The rest of this sub-section describes more details of the two corpora including data sizes and the acoustic properties of videos.

How2 dataset

The How2 dataset consists of instruction videos collected from YouTube (Sanabria et al., 2018). Each video is accompanied by a user-uploaded closed caption. The offi-

[2] https://www.udacity.com/

cial partition of the corpus is 512 hours of training data, 5.5 hours of validation data and 4.7 hours of test data. The corpus contains 19,804 videos with video titles for each one. The vocabulary size of the video collection calculated from user-uploaded closed captions is roughly 38,600. The speaking style of How2 videos is monologue with speakers either indoors or outdoors. In outdoor locations, background noise can be present in a video which can impact on speech recognition accuracy.

Udacity dataset

The Udacity dataset is a collection of online lecture videos downloaded from the Udacity website. The videos contained in the dataset were downloaded by the author of this thesis. In the corpus, 60 courses on engineering, science, programming topics are available. For the ASR experiments, the dataset was partitioned into 50 courses used for ASR training and 5 courses each for validation and test set, respectively. Although the same speakers can appear in multiple courses, the dataset are carefully partitioned to ensure that speakers in the training and validation data are not present in test set. The duration of the training data is 163 hours, that of the validation and test data are approximately 3.5 hours each. The closed captions corresponding to the videos were also downloaded. The vocabulary size of the closed captions of all 60 courses is roughly 21,600. Speech contained in this corpus is mostly monologue, but a few videos show short skits involving multiple speakers having conversations.

5.2.2 Multimodal feature extraction

An important data pre-processing stage for multimodal augmentation of ASR systems is multimodal feature extraction. The three multimodal features explored in this chapter are the visual context feature, the video title feature and the speaker face feature. While the visual context feature and the speaker face feature are extracted using pre-trained computer vision models as feature extractors, the video title feature is produced by transforming word embedding vectors corresponding to words in a video title. This sub-section describes details of how these three features are prepared for our experiments.

Visual context feature

The visual context feature was extracted using a pre-trained object recognition model as described in Section 5.1.1. This investigation used the VGG19 model pre-trained for the object recognition task on ImageNet data (Simonyan and Zisserman, 2015; Russakovsky et al., 2015). The VGG19 model consists of 19 layers of convolutional operations and the pre-traind model can classify an input image into 1,000 pre-defined categories of ImageNet data. The pre-defined categories include animals ("terrier", "jellyfish", "koala"), objects ("baseball", "iPod", "monitor"), and food ("ice cream", "carbonara", "pizza"). The visual context feature was an output vector of the size 4,096 extracted from the penultimate layer of the model given input. For each speech utterance, an input image of the visual context feature extractor was taken from the middle frame of video. This middle frame of an utterance was assumed to be a representative of the utterance, and the same approach was taken in previous work (Gupta et al., 2017). Although no formal evaluation of extraction of representative video frames was conducted, this approach seemed sufficient when randomly selected around 100 sample video frames were manually examined.

Video title feature

The video title feature is an abstract representation of a short summary representing each video. A video title feature was either the average or sum of word embedding vectors of video title tokens (Section 5.1.1). Comparison of these two approaches is described in Section 5.3.1. Another variable of generation of a video title feature was a training method for a word embedding model. The RNN language model as shown in Figure 5.1 incorporates a word embedding model to transform input word tokens into word embedding vectors. The first approach to generating word embedding vectors for the video title feature was to use the word embedding model incorporated in the RNN language model and share the model for both input word embedding vectors of the RNN language model and input word embedding vectors of the video title feature. The second approach was to train another word embedding model independent of the model incorporated into the RNN language model. These two approaches to developing word embedding models for creation of the video title feature are also examined in Section 5.3.1. For development of the word embedding model independent of the RNN language model, the fastText library (Bojanowski et al., 2016) was used and the model was trained on train data of How2 for the How2 experiments and on that of Udacity for the Udacity experiments. In both How2 and Udacity datasets, several utterances could be derived from a single video. When multiple utterances belonged to the same video, these utterances shared the video title feature derived from the video.

Speaker face feature

The speaker face feature was employed to provide an neural acoustic model with visual demographic information contained in a vector representation of a speaker face. Extraction of the speaker face feature was carried out in the two stages. The first stage was to detect a speaker face in a video using a face detector and to generate face tracks. The second stage was to apply a face recognition model to the extracted face tracks and aggregate features from each face track to produce the final speaker face feature.

The first stage of extraction of the speaker face feature was to apply a face detector to videos. For each video, output of the face detector was a list of face tracks found. We applied a short boundary detector before using a face detector to reduce the number of frames to which the face detector was applied.

1. a shot boundary detector segments a video based on transitions of different visual patterns.

2. face detection and tracker identifies human faces present in detected shots.

Face detection was computationally demanding and face tracking was applied to every 10 seconds of each detected shot, so that this avoided applying a face detector to every single frame in each shot. This could result in failure to detect a face when this stride was too large. For videos where no face was detected at initial run of a face detector, the face detector was applied to every single frame of each shot. When a speaker face was still not detected due to failure of the face detector or absence of a speaker image in a video, the speaker face feature was substituted for a zero vector.

The second stage of extraction of the speaker face was to apply a pre-trained face recognition model to a list of face tracks extracted in the first stage. The face recognition model used as an embedding extractor was a residual network architecture trained on roughly 3 million faces for classification of 7,485 people as target (He et al., 2016). The model was pre-trained to classify faces of popular figures including politicians, actors and sports athletes. The pre-trained model was available as part of the dlib library (King, 2009). The experiments on the neural acoustic model adaptation were conducted using the How2 data. Since the How2 data contained one speaker per video, the face recognition model was applied to the longest face tracks detected in the first stage, assuming that the tracks corresponded to a speaker face. The extracted embedding vectors from face tracks were aggregated by taking the average of all embedding vectors to form a speaker face feature. As shown in analysis of Section 5.4.2, this approach was sufficient for an neural acoustic model to learn a speaker gender from a visual feature.

5.2.3 ASR architecture

This sub-section describes the baseline system used for all experiments and technical details of multimodal models including hyper-parameters and other configurations.

Baseline system

As described in Section 2.1, a standard hybrid ASR architecture consists of an acoustic model and an N-gram language model. For our experiments on the use of a multimodal RNN language model and a multimodal neural acoustic model, hybrid ASR systems were built using the Kaldi ASR toolkit (Povey et al., 2011). The details of the baseline system without multimodal features are as follows. For all experiments, the acoustic features were a 40 dimensional filter bank with a window length of 25ms and 10ms frame shift. For all the experiments, the acoustic model consisted of 6 layers of time-delayed architecture which had a tapered shape towards the output layer more suitable to model acoustic features (Peddinti et al., 2015). The N-gram language model was a 3-gram using modified Kneser-Ney smoothing with the SRILM toolkit (Chen and Goodman, 1995; Stolcke, 2002).

Multimodal RNN language model

The ASR systems for the experiments on the multimodal RNN language model were developed with the configuration described above. For experiments on the multimodal RNN language model, one RNN language model was trained on How2 closed captions of train data and another RNN language model was trained on Udacity closed captions of train data. For each dataset, 30-best hypotheses were generated from decoded lattices created by first decoding of utterances using an acoustic model and an N-gram language model (Section 2.1.4). To examine contributions of the visual context feature and the video title feature, the RNN language models with and without the multimodal features were used to re-score these 30-best hypotheses. The number of N-best 30 was previously successfully used for an investigation by Gupta et al. (2017).

The architecture of the RNN language models consisted of two RNN layers with LSTM cells of 512 hidden sizes (Section 2.1.3). In this experiment, the size of 512 was better than 256 and 1,024. The size of word embedding vectors were set to 100 following (Gupta et al., 2017). The RNN language models were trained to predict the next word w_{i+1} using the cross-entropy loss where given a current word input w_i and the hidden state from the previous input h_{i-1}. The learning rate was set to 20 and divided by 4, when no improvement of perplexity was observed on the validation set after each training epoch. These values were adopted from an example script of an RNN language model[3]. The models were trained for 50 epochs. Dropout was applied to the RNN language models with the rate of 0.2 to avoid model overfitting on train data. A mini-batch size of 100 utterances were used for the Udacity model, while 90 utterances were used for the How2 model, because these numbers were the maximum capacity that the GPUs used for training had. The RNN language models were trained using the PyTorch deep learning library (Paszke et al., 2017).

Multimodal neural acoustic model

The experiments on the multimodal neural acoustic model were only conducted on the How2 corpus. The Udacity corpus was not suitable for development of the multimodal neural acoustic model using a speaker face feature because the videos contained many slides in a visual stream and multiple speakers could be present in a video. For this reason, a multimodal ASR system was trained on How2 data. Due to the intensive computational requirements to produce speaker face features, the only a randomly selected subset of the How2 corresponding to 107 hours of videos were used for training of the multimodal acoustic model. The N-gram language model was trained on all text of the How2 training data, that is the identical N-gram language model training set to the previous experiment. The size of speaker face features was 128 and combined with the 40-dimensional acoustic filter bank features, input of the multimodal neural acoustic model was 168 when the speaker face feature was used.

For experiments, four different neural acoustic models were developed to examine the effects of the speaker face feature with regard to the WER. There was a baseline system without any additional feature, a system with an i-vector feature (Section 2.3.2) (Dehak et al., 2011), a system with a speaker face feature and a system with combination of an i-vector feature with a speaker face feature. An i-vector extractor used was a Gaussian mixture model of 1,024 components and a total variability matrix was trained on regions of How2 audio where voice activity

[3] `https://github.com/pytorch/examples/blob/main/word_language_model/main.py`

detection identified speech. The size of the i-vector feature was 128. These were default parameters of i-vector training of Kaldi script. Assuming that each How2 video contains one speaker, speaker i-vectors were computed by taking the average of all the utterances of each video.

5.3 Experimental Results

Previous sections described our approaches to integration of the three multimodal features, the visual context feature, the video title feature and the speaker face feature into an ASR system. While the multimodal RNN language models will be used for ASR N-best re-scoring, the multimodal neural acoustic model is directly applied to decoding speech utterances. Experiments conducted in this section examine the effects of the three multimodal features with regard to the WER metrics. This section first shows experimental results of the multimodal RNN language models using the visual context feature and the video title feature. Following the language model experiments, the WER results of the multimodal neural acoustic model using the speaker face feature are presented. Unless specified, changes in evaluation metrics reported here are absolute changes.

5.3.1 Results on the RNN language model and multimodal augmentation

This section provides WER results of the RNN language models. The first set of results demonstrates four different configurations discussed in Section 5.1.1. These experiments explored the effects of (i) averaging vs summation of word embedding vectors for creation of the video title feature, and (ii) the use of a word embedding model integrated into the RNN language model vs a word embedding model independent of the RNN language model using the fastText library. The second set of experimental results demonstrate the effects of the visual context feature and the video title feature for ASR N-best re-scoring.

Approaches to creating the video title feature

The first sub-section shows experimental results on different approaches to creation of the video title feature. As discussed in Section 5.1.1, four different configurations could be considered when creating the video title feature. The following shows use of averaging or summation for transforming word embedding vectors of video title tokens into a single vector representation.

- **sum** take summation of word embedding vectors of video title tokens
- **ave** take average of word embedding vectors of video title tokens

Creation of a word embedding model could have two options.

- **shared** use a word embedding model integrated into the RNN language model for both creation of input token representations **and** for creation of token representations of video title words
- **fastText** build a word embedding model separately from the integrated word embedding model using an external library (this experiment used the fastText library)

The perplexity and WER results of using different approaches to creating the video title feature are summarised in Table 5.1. The WERs were obtained by applying the RNN language model with the video title feature to ASR N-best re-scoring. Generally both perplexity and WER values were better when the video title feature was created from a word embedding model independent of the one integrated into the RNN language model (*fastText* systems). There was not a big gap observed between averaging and summation of word embedding vectors of a video title. When using the video title feature, therefore, it is recommended to train a separate word embedding model for producing word embedding vectors of video title words.

Table 5.1: Comparative results of different methods to produce the video title feature. PPL is perplexity, and WER is word error rate.

	Udacity		How2	
	PPL	WER	PPL	WER
shared, sum	138.50	14.67	63.21	18.33
shared, ave	138.87	14.70	64.12	18.44
fastText, sum	129.07	14.49	61.16	18.40
fastText, ave	131.69	14.48	60.03	18.27

WER results of using visual context and video title features

The second set of experiments investigated the effects of the multimodal features used to augment RNN language models. There are five different systems compared in these experiments.

- **no-rescoring** Direct output of decode without applying RNN re-scoring
- **vanilla** The baseline RNN language model without any multimodal feature
- **visual_context** The RNN language model with the visual context feature
- **video_title** The RNN language model with the video title feature
- **combined** The average of scores from the N-gram, the vanilla RNN language model, the visual context model and the video title model

Table 5.2 summarises perplexity and WER results of using the vanilla RNN language model and the RNN language models using the multimodal features. Compared to the vanilla model, both the visual context feature and the video title feature led to reduction in perplexity. On the Udacity corpus, both multimodal features reduced perplexity roughly by 5. On the How2 corpus, perplexity was improved by 5 using the visual context feature and by more than 10 using the video title feature.

Improvement of the WERs using the multimodal features was, however, marginal or negative. Neither the visual context nor the video title features reduced WERs on the Udacity corpus. On the other hand, the visual context feature reduced WER by 0.12% and the video title feature reduced WER by 0.29% on the How2 corpus.

Table 5.2: Experimental results of the RNN language models using the multimodal features. “oracle” is the best achievable WER of the after N-best re-scorings.

	Udacity		How2	
	PPL	WER	PPL	WER
no-rescoring	-	16.70	-	20.25
vanilla	136.00	14.50	71.43	18.56
visual_context	131.29	14.71	65.79	18.43
video_title	131.69	14.48	60.03	18.27
combined	-	13.97	-	17.98
oracle		9.20	-	15.13

When scores from the N-gram, vanilla, visual feature and video title models were interpolated to re-score hypotheses (Section 5.1.1), the WER was further reduced by 0.53% and 0.58% on Udacity and How2, respectively.

Overall, both the visual context and video title features contributed to consistent improvement in perplexity on the both corpora. However, the multimodal features did not lead to reduction in WERs on the Udacity corpus. A possible explanation for this is that Udacity videos present lecture slides (e.g., characters, equations, flowcharts, and programming code), or humans (e.g., lecturers, and interviews), which were not part of the ImageNet dataset. Another cause could be that the number of video titles used on the Udacity was small (i.e., 50), while there were more than 19,000 video titles used for the How2 model. In addition, the Udacity video titles tended to be shorter and could have been less informative. For example, one video title of the How2 corpus was “Intermediate Western Calligraphy Tips: The Acanthus Leaf & Calligraphy: Part 1”, while the Udacity corpus has a short generic video title “How to Build a Startup”.

5.3.2 Results on the multimodal acoustic model

The experiments on the multimodal neural acoustic model were carried out using the How2 corpus. As mentioned earlier, the Udacity corpus was not suitable for experiments due to content mainly containing lecture slides and absence of speaker

Table 5.3: WER results for the multimodal neural acoustic model on the How2 corpus.

	WER
baseline	22.65
speaker_face	22.48
i-vector	22.66
i-vector+speaker_face	22.60

faces in most parts of videos. There were four different systems built for the experiments. The experiments were carried out on a subset of the How2 corpus due to the intense computation cost of detection and extraction of speaker faces (Section 5.2.3).

- **baseline** baseline neural acoustic model without using extra features
- **speaker_face** acoustic model using the speaker face feature
- **ivector** acoustic model using the i-vector feature
- **ivector+speaker_face** acoustic model using both the i-vector and speaker face features

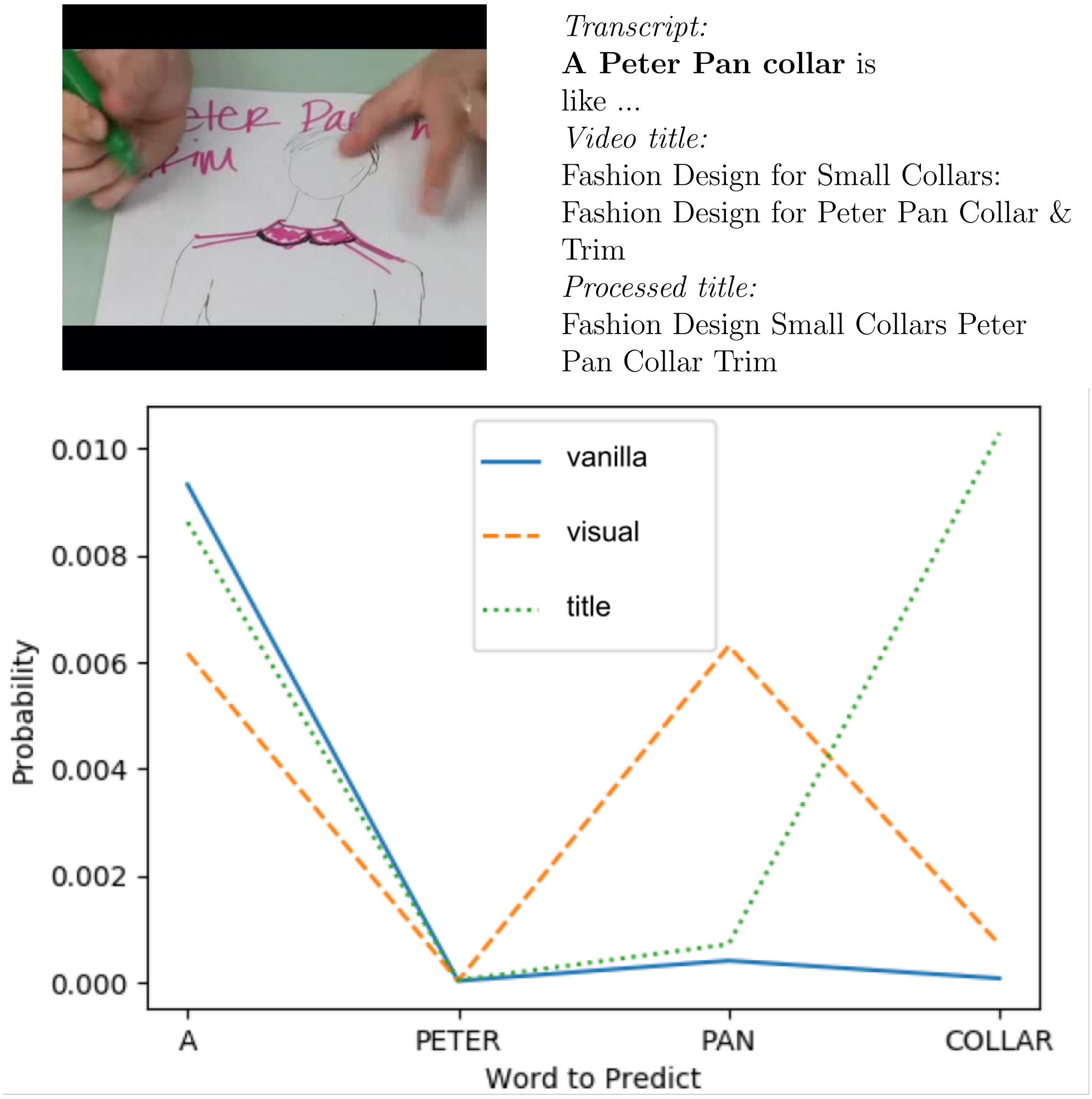
Table 5.3 shows the WER results of neural acoustic model adaptation using the speaker face feature and the i-vector feature. These WERs in the table were higher than the ones on Table 5.2 due to training acoustic models on less amount of data. Adding the speaker face feature for adaptation led to a small gain in WER by 0.17%. Although i-vector has been demonstrated to contribute to WER improvement (Saon et al., 2013), in this investigation using roughly 100 hours of How2 training data did not lead to improvement. Combination of the i-vector with the speaker face feature was not effective either.

5.4 Model analysis

The experimental results in Section 5.3 showed marginal improvement of WERs using the three multimodal feautures investigated in this thesis. Although WER gain was small, it is interesting to examine whether multimodally augmented RNN language models and neural acoustic models learned information from non-speech sources. This section provides an analysis of the impact of multimodal features on these models. Analysis of RNN language models investigates whether predictions of words were conditioned on the visual context and video title features. The augmented models are expected to predict content words (e.g., nouns) with a higher probability thanks to the multimodal feature carrying the situational and background contexts of speech. Analysis of the neural acoustic model using the speaker face feature examines whether recognition accuracy is affected when the speaker face feature taken from irrelevant speaker is used. The hypothesis of this analysis is that the multimodally augmented acoustic model is aware of a relationship between a visual speaker face feature and speech characteristics and this causes drop in WERs when a mismatch between speech and the speaker face feature occurs.

5.4.1 Analysis of the multimodal RNN language model

The RNN language model conditions its prediction of the next word on the preceding contexts observed. The multimodal RNN language model additionally conditions its prediction of the next word on the situation contexts contained in the visual context feature and the video title feature. The multimodal feature should therefore give higher confidence to the RNN language model when determining probabilities of the two acoustically similar but semantically different phrases. For example, when the multimodal feature is taken from a visual image of sand and a beach, the associated speech likely to be spoken is “wreck a nice beach” rather than “recognise speech”. To further analyse the effects of multimodal features on the RNN language model, one utterance from the Udacity corpus and one utterance from the How2 were randomly selected to show illustrative examples of an image, a video title and word probabilities from the model. Further, 10 “keywords” from Udacity and How2 were randomly selected to examine how well the multimodal language models predicted these “keywords”.



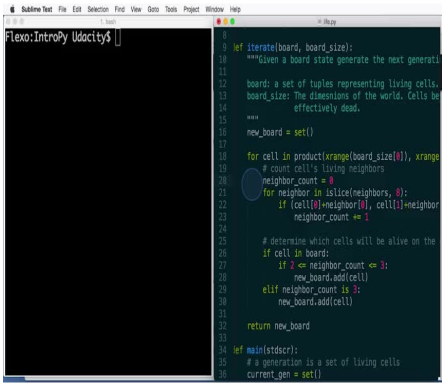
Transcript:
A Peter Pan collar is like ...
Video title:
Fashion Design for Small Collars: Fashion Design for Peter Pan Collar & Trim
Processed title:
Fashion Design Small Collars Peter Pan Collar Trim

Figure 5.3: An excerpt of a How2 utterance transcript, the corresponding video title and video frame. The graphs show probabilities of words in the utterance transcript produced by three different language models: vanilla, visual feature ("visual") and video title ("title").

Figure 5.3 shows word probabilities of an excerpt of a How2 utterance. The individual line graphs show probabilities of the vanilla language model, the visual context language model and the video title language model. This example demonstrates that multimodally augmented language models gave a higher probability to "Pan" and "collar" than the vanilla model. The corresponding image contains a collar-like object and both "Pan" and "collar" are included in its video title. This indicates that the multimodal language models figured out a relationship between a visual collar-like object and "collar" uttered in speech.

Figure 5.4 shows analysis of an utterance taken from the Udacity corpus. The video frame of this utterance shows coding blocks of Python and the video title contains the term "Python" and "programming". As can be seen in the graphs, the probability of "Python" produced by the multimodal language models is higher than the vanilla model. This indicates that the video titles could be useful for the neural language model to understand that the video content covers a topic of Python programming, while the visual context feature helps the model to know that the current utterance is relevant to programming or demonstration of computer language interpreter.

The second analysis on the multimodal language models examines word probabilities produced for prediction of 10 content words. For example, if the word "cat" appears in a corpus 3 times and the language model produces probabilities 40%, 50%, 30%, the average probability of the "cat" in this corpus is 40%. Table 5.4 summarises the average probabilities of 5 words on test set of How2 and 5 words on test set of the Udacity corpus. Each noun is one of the keywords of a video. For example, most of the "testing" on the test set of the Udacity corpus occurs in the course "AB Testing for Analysis Business", and many examples of "nose" on the test set of the How-to corpus are found in "Cosmetics: Narrow the Nose with Makeup". The multimodal models were generally better at prediction of keywords randomly selected from the two corpora. The video title model was especially good at giving higher confidence in prediction of the nouns, except for "python" and



```
Flexo:IntroPy Udacity$ 
```

```
lef iterate(board, board_size):
    """Given a board state generate the next generati

    board: a set of tuples representing living cells.
    board_size: The dimesnions of the world. Cells be
                effectively dead.
    """
    new_board = set()

    for cell in product(xrange(board_size[0]), xrange
        # count cell's living neighbors
        neighbor_count = 0
        for neighbor in islice(neighbors, 8):
            if (cell[0]+neighbor[0], cell[1]+neighbor
                neighbor_count += 1

        # determine which cells will be alive on the
        if cell in board:
            if 2 <= neighbor_count <= 3:
                new_board.add(cell)
        elif neighbor_count is 3:
            new_board.add(cell)

    return new_board

lef main(stdscr):
    # a generation is a set of living cells
    current_gen = set()
```

Transcript:
Indentation levels in Python are always ...
Video title:
Introduction to Python Programming
Processed title:
Introduction Python Programming

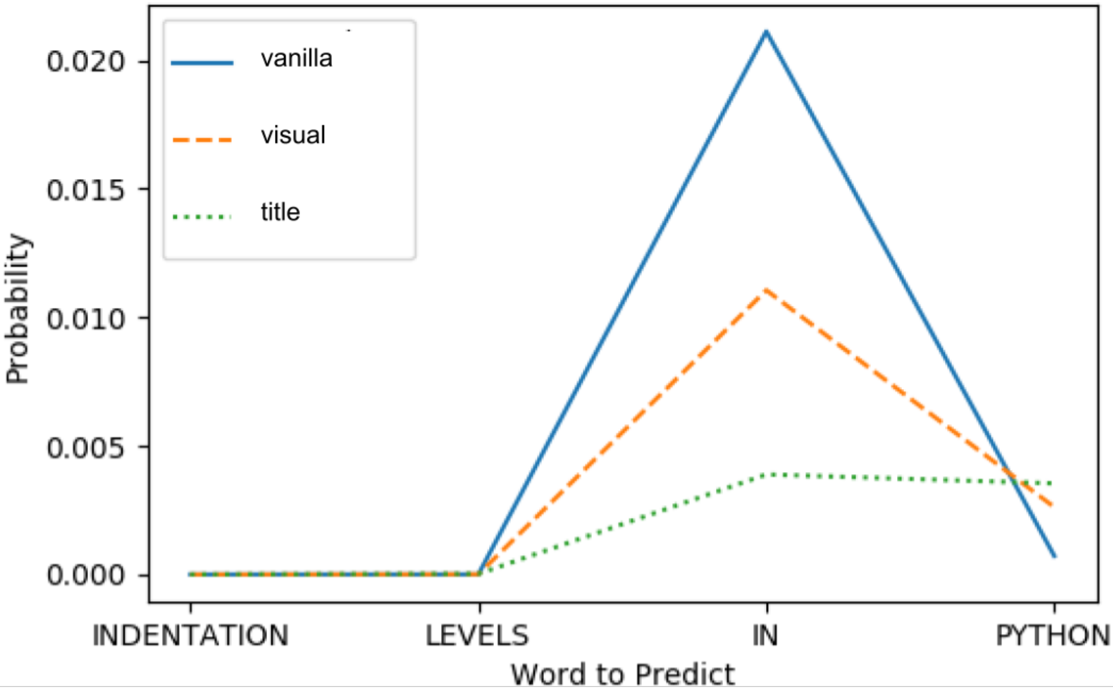


Figure 5.4: An excerpt of an Udacity utterance transcript, the corresponding video title and video frame. The graphs show probabilities of words in the utterance transcript produced by three different language models: vanilla, visual feature ("visual") and video title ("title").

"chord" in comparison to the vanilla model. The visual context model produced a higher or comparable probability for all but "python", "collar" and "golf" than the vanilla model. This analysis might encourage further research on the use of word probabilities from multimodal language models for SCR tasks.

Table 5.4: Average word probabilities of 10 noun "keywords" produced by the RNN language models. The probabilities were averaged by the total word frequencies on the test set. The train column shows frequencies of the words on the training split of the corpora, and the test column on the test split of the corpora

	probabilities (%)			frequency (#)	
	vanilla	visual	title	train	test
Udacity					
ad	0.06	0.04	0.37	107	205
analysis	0.08	0.34	0.19	263	47
python	0.44	0.19	0.41	467	43
readme	0.004	0.01	0.04	20	17
testing	0.18	1.33	1.73	367	12
How-to					
chord	21.56	20.86	20.82	495	10
collar	0.55	0.18	0.85	245	5
fish	2.34	6.07	14.49	812	36
golf	0.55	0.39	1.42	346	8
nose	0.6	5.01	6.85	752	11

5.4.2 Analysis of the speaker face feature

The influence of the speaker face feature on the neural acoustic model was analysed by deliberately creating a mismatch between speech and the corresponding speaker face feature. The hypothesis here is that if the multimodal acoustic model learned the relationship between speech and a visual speaker face, the WER would become worse with a mismatched speaker face feature. To analyse whether the multimodal acoustic model learned the relationship between speaker demographic information with speech, 6 videos from the How2 test set of 3 male speakers and 3 female speakers were randomly selected. The choice of speaker gender was merely to analyse the

Table 5.5: WER results of the multimodal acoustic model using the mismatched scenario of the speaker face feature. Each of the 6 speaker face features was used to compute video-wise WERs. Videos labelled with f contain a female speaker and with m contain a male speaker.

		Speaker face					
		f1	f2	f3	m1	m2	m3
Speaker	f1	6.39	**5.17**	6.71	8.29	9.51	6.9
	f2	**14.59**	16.33	16.06	16.5	20.19	18.64
	f3	20.51	20.62	**20.09**	25.07	25.84	25.22
	m1	31.42	27.39	32.63	24.28	**24.0**	26.14
	m2	32.26	31.27	32.68	**23.77**	25.49	26.61
	m3	40.24	47.09	38.07	**23.08**	24.60	31.89

effects of demographic information contained in a visual stream for the acoustic model but any other attributes could be considered. For each video, the 6 different speaker face features were provided to the acoustic model to decode utterances of 6 videos. Table 5.5 summarises WER results of this experiment. As can be observed in the table, an increase in WER is seen when the gender of the speaker face feature was mismatched with the actual gender of a speaker. For example, the WERs of the video of the third female speaker "f3" were around 20% when the speaker face feature was also taken from a female speaker, whereas WERs increased to around 25% with the speaker face feature of a male speaker. In the speaker face feature, information about speaker gender was not explicitly embedded. This analysis indicates that the acoustic model might be able to learn the relationship between visual speaker face and speech, and future research on this direction could illuminate information which the acoustic model learned from the provided visual features.

5.5 Conclusions and Future Work

This chapter investigated integration of multimodal features into the hybrid ASR system. The use of multimodal information for ASR is motivated by the fact that human language understanding exploits situational contexts observed in the real world surrounding speakers. Thanks to advances in computer vision technology, various pre-trained computer vision models are currently available which can be used as a visual feature extractor. This investigation examined the use of three different multimodal features: (i) the visual context feature, (ii) the video title feature and (iii) the speaker face feature. The visual context feature and the video title feature were used to augment the neural language model. The ASR N-best hypotheses were generated from decoder lattices and the multimodally augmented language model were used to re-score N-best hypotheses. The speaker face feature was directly fed to an acoustic model and provided the model with demographic information about the speaker. However, experimental results showed that the multimodal features investigated in this chapter brought only a marginal improvement of WERs. Analysis of the effects of visual context and video title features on the RNN language model shows that the multimodal language model predicted some randomly chosen "keywords" with higher confidence. Analysis of the effects of visual speaker features on the acoustic model suggests that the acoustic model might be able to learn speaker attributes as in the form of speaker gender in this analysis from visual information. These analyses can encourage future research on what acoustic and language models could learn from multimodal information.

With respect to SCR research, however, these multimodal extensions of ASR systems can have a little impact on search effectiveness. Neither the multimodally augmented acoustic and language models brought less than 1% WER improvement. As discussed Section 2.1.5, the relationship between WER and search effectiveness is near linear (Garofolo et al., 2000) and this degree of improvement is unlikely to have an influence on ASR transcripts for search models.

Following our investigation of the use of multimodal features in ASR, in the next chapter, we explore semi-supervised training and content genre adaptation for ASR. The use of these techniques are motivated to improve transcription accuracy of highly varied spoken document collections. The next chapter introduces the pipeline architecture of semi-supervised training and our approach to acoustic model adaptation using content genre. Our experimental results show a gain in recognition accuracy using these techniques.

Chapter 6

Semi-supervised Training and Acoustic Model Adaptation using Content Genre for ASR

As described in Chapter 1, state-of-the-art ASR systems can still produce significant numbers of word errors on highly varied content. This issue is avoided in many ASR settings where transcription are required in a specific domain on that area and the ASR system is developed or adapted for the specific domain. However, SCR generally needs to operate on Spoken Content which spans broader topic area. This is exemplified by user-generated content which is often highly varied. WERs for this content can reach 30-40% which as outcome can hinder search efficiency (Larson and Jones, 2012). It is therefore desirable to improve the accuracy of ASR systems for such multi-domain spoken content where the topics coverage of the content cannot be controlled or predicted. In this chapter we examine methods which may improve ASR for multi-domain content.

Two potential approaches to achieving improved multi-domain ASR are investigated in this chapter: (i) exploiting untranscribed speech data with semi-supervised methods (Manohar et al., 2018) and (ii) use of domain adaptation techniques using for ASR (Abdel-Hamid and Jiang, 2013; Saon et al., 2013) using user-provided genre

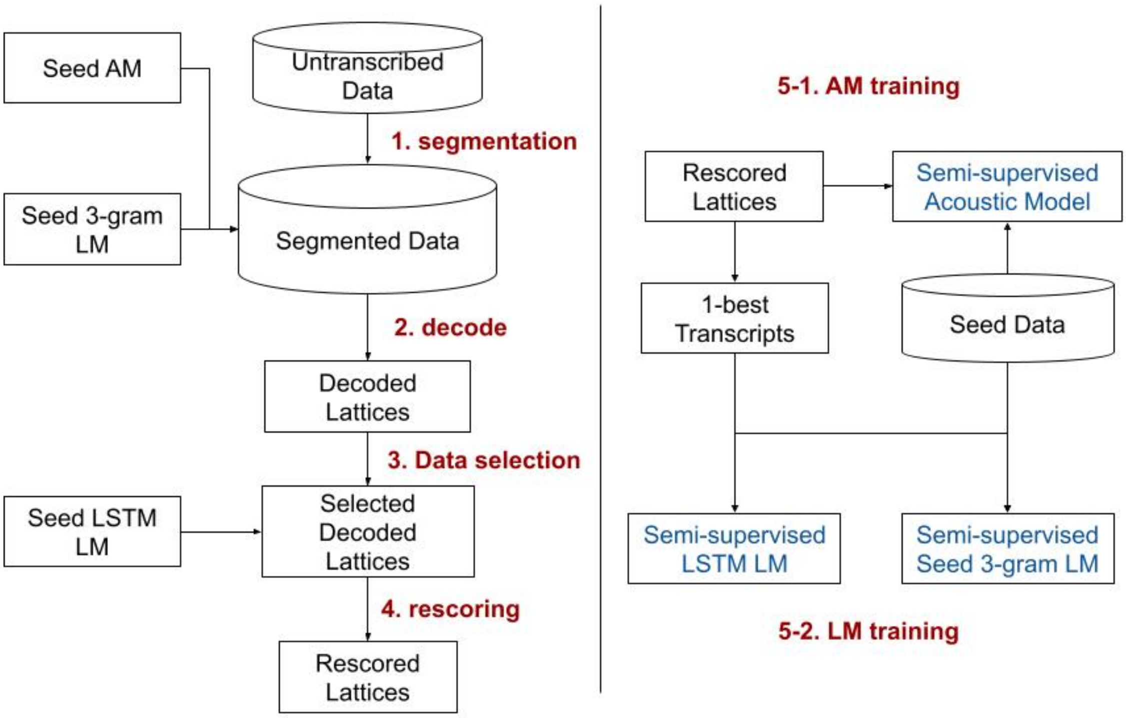


Figure 6.1: Flowchart of semi-supervised training for acoustic and language models.

tags in a corpus. In this thesis, the use of the term "genre" is limited to classification of spoken content provided by content creators (e.g., "yorkshireterrier" is not a genre tag but "conference" is), while the term "domain" is used for general categorisation of spoken content. The overall goal of this investigation is to establish ASR methods which can improve SCR effectiveness. This chapter reports WER gain achieved with semi-supervised training and domain adaptation in ASR. Results of an SCR investigation using these transcripts appear in Chapter 7. This work was accepted at the 2021 Statistical Language and Speech Processing conference[1].

6.1 Semi-supervised training for ASR

Semi-supervised training for ASR exploits untranscribed data as additional data which is usually in a domain not covered in training data (Manohar et al., 2018; Veselỳ et al., 2013). For example, a large amount of manual labels for low re-

[1] My contribution is background research, designing and conducting experiments and analysis of the experimental results.

source languages are often not available for ASR training data and untranscribed data for a target language can be exploited for additional training data resources (Carmantini et al., 2019; Su and Xu, 2015; Veselỳ et al., 2013). Existing work on semi-supervised training has shown WER gain in 7% when using the conversational telephone speech of the laboratory corpus and WER gain in 3% in the low resource settings of Vietnamese speech recognition.

In a similar vein, covering manual labels for broad topic areas for multi-domain ASR is costly and time-consuming. Existing work on semi-supervised training has not examined the semi-supervised approach to potential improvement of multi-domain ASR. Untranscribed spoken data found in the wild is often not cleaned or segmented, unlike ASR data in the laboratory settings. We hypothesise that optimising data segmentation, data selection, acoustic modelling and language modelling in the contexts of semi-supervised training can reduce WERs for multi-domain content. Figure 6.1 shows a flowchart for the application of the semi-supervised approach to acoustic model and language model training in ASR. The rest of this section shows each step involved in semi-supervised training for multi-domain ASR.

Data segmentation

Natural audio generally consists of a mixture of speech utterances and other audio activities. The curated speech corpora used in existing ASR research are typically pre-segmented into speech utterances (Panayotov et al., 2015; Sanabria et al., 2018). However, user-generated spoken content is typically not pre-segmented into speech utterances and regions of speech are unknown. Since it is unrealistic to create manual labelling of speech and non-speech regions for such data, we need to seek to do this labelling automatically. In this investigation we examine the use of voice activity detection (VAD) which seeks to detect regions of active speech. This is based on the observation that we should only require the ASR system to transcribe spoken content. However, the semi-supervised training process may be sufficiently robust to work effectively with a mixture of speech and other audio events. To examine this hypothesis we also investigate a simple method which does not attempt to distinguish between speech and non-speech audio. Equal segmentation divides each audio file into fixed duration regions of audio regardless of audio content. WER results comparing equal segments with VAD are presented in Section 6.4.

Data selection

Data selection is used to select segments of untranscribed data which are likely to lead to sufficiently accurate ASR for semi-supervised training of an acoustic model and a language model. This investigation employs the segment level confidence score described in (Yu et al., 2010). The segment level confidence score can be computed by taking the average of posterior probabilities of speech segments decoded by a seed ASR system. Our experiments demonstrate application of the confidence score to equal sized and VAD determined speech segments and its effect on WER.

Acoustic model

For acoustic model training using untranscribed data, the recently proposed training method proposed by Manohar et al. (2018), semi-supervised lattice-free maximum mutual information (LF-MMI) is employed. As described in Section 2.3.1, semi-supervised LF-MMI is a sequence discriminative training method which exploits lattices of decoded untranscribed data for training of a new acoustic model. This method has been demonstrated to be better than training a new system on standard 1-best hypotheses of untranscribed data (Manohar et al., 2018).

Language model

As shown in Figure 6.1, a seed N-gram language model is used to decode untranscribed data and a seed RNN language model is used to re-score decoded lattices of untranscribed data (Xu et al., 2018). These language models are trained on manual transcripts of a speech corpus. For our investigation, we generate a 1-best transcript of the untranscribed data, and train new N-gram and RNN language models on a combination of manual transcripts of the seed data with ASR transcripts of untranscribed data. We examine the benefits of incorporating ASR transcripts from varied domains in language model training. While Çelebi and Saraçlar (2013) investigated the use of untranscribed data for an N-gram language model, semi-supervised training of an RNN language model has not been explored.

6.2 Acoustic model adaptation using genre ID and genre embedding

Domain adaptation is a technique to provide an acoustic model with information that is useful to recognise speech of a target domain. Existing work has explored adaptation of a neural acoustic model using speaker specific information called "i-vector" features (Saon et al., 2013) and acoustic environmental information (e.g., "kitchen", "cafeteria") (Fainberg et al., 2017). Adaptation using an "i-vector" is a widely used approach to acoustic DNN adaptation bringing 1-2% absolute WER improvement (Section 2.3.2) (Saon et al., 2013). While the i-vector feature is created using a GMM model and factor analysis, extraction of a speaker specific feature is further improved by using a DNN feature extractor, giving features called "x-vectors" (Snyder et al., 2018). Acoustic models using an "x-vector" improve WER gain by around 2% over the "i-vector" system (Snyder et al., 2018). To build an x-vector extractor, a DNN model is trained to classify an acoustic feature vector into a pre-defined speaker label. The trained DNN model captures a relationship between acoustic features and speaker information and a feature vector extracted from the DNN model represents information about the speaker.

As outlined at the beginning of this chapter, an SCR system can encounter a broad range of topics in a large content archive. To enhance recognition accuracy of multi-domain content, our investigation examines the use of genre tag for adaptation of a neural acoustic model. User-provided tags classifying content are often available as part of metadata of content from content creators. For example, YouTube videos can be searched by a tag prefixed with a hash symbol in a description field (e.g., "#yorkshireterrier", "#harpseal") and Spotify podcasts are classified into categories

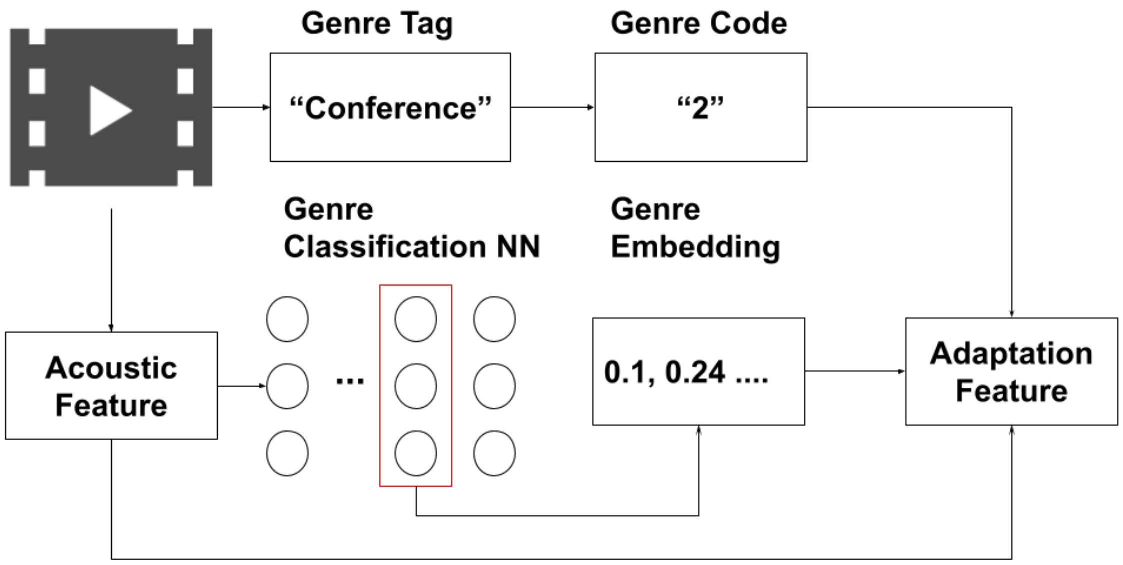


Figure 6.2: Generation steps of genre code and genre embedding for content genre adaptation of an acoustic model.

such as educational and technology. We hypothesise that user-provided tags showing content genre is an indicator of acoustic noise contained in the content, and that a neural acoustic model can be more robust to a certain type of noise (e.g., "applause") that appears more often in a certain genre tag (e.g., "conference").

We investigate two approaches to adaptation of a neural acoustic model using genre tags. Figure 6.2 shows creation of adaptation features using genre information of spoken content. These approaches assume that genre tags are provided by content creators. In our experiment, videos of the Blip10000 corpus (Schmiedeke et al., 2013) are accompanied by 26 different genre tags uploaded by creators. The first approach is to use a genre tag as a single digit code (genre code) and append it to an acoustic feature vector. This is similar to the domain ID used in (Sainath et al., 2020), however, our genre codes are employed in the semi-supervised settings. The second approach is to build a genre embedding extractor using a DNN similar to the x-vector extractor (Snyder et al., 2018) and to feed an extracted genre embedding vector to a neural acoustic model. The genre embedding extractor is trained to classify an input acoustic feature vector into a pre-defined set of genre labels. The extractor is hypothesised to learn a relationship between acoustic noise present in certain genres and genre tags and the embedding vector can enable an acoustic model to be robust to acoustic noise likely to be present in a certain genre tag. These adaptation features are concatenated with an acoustic feature vector to be the input to a neural acoustic model.

6.3 Experimental setup

This section describes the experimental setup for our investigation of semi-supervised training and acoustic model adaptation for multi-domain ASR. For these experiments we use the Blip10000 corpus (Schmiedeke et al., 2013), a collection of highly varied user generated videos. This section is structured as follows. We begin with a description of the Blip10000 corpus, and present details of the baseline model architecture and model training. This description includes details of equal and VAD segmentation for audio, and extraction of genre codes and genre embedding.

6.3.1 Datasets

Semi-supervised training requires an existing ASR system, referred to as "seed system", trained on out-of-domain data to be available. For our study, two seed ASR systems were trained on publicly available corpora How2 (Sanabria et al., 2018) and LibriSpeech (Panayotov et al., 2015). The How2 corpus was introduced in Section 5.2.1. The Blip10000 corpus contains user-generated videos of diverse qualities and genres crawled from the Internet. Following the overview of LibriSpeech, details of the Blip10000 corpus and creation of manual transcripts for the corpus are presented.

Librispeech

The LibriSpeech corpus is a collection of 1,000 hours of English audio-books (Panayotov et al., 2015). The size of corpus is 1,000 hours. The corpus is accompanied by manual transcripts and speech is carefully segmented into speech utterances. The corpus contains two different setups for system evaluation; a test set of clean speech and a test set of noisy speech. There are several partitions available {100, 360, 460, 960} hours. We used the largest 960 hours partition for experiments.

Blip10000

The Blip10000 corpus consists of 14,838 user-generated videos (3,288 hours) collected from the Internet and is released under Creative Commons. The corpus is partitioned into 5,288 videos for a dev set and 9,550 videos for a test set. The Blip10000 corpus contains videos of 26 different genres; its content includes materials such as vlogs, conferences, street interviews, semi-professional broadcasts, technology reviews and so on. The spoken language is mainly English, but non-English videos can be found. To use blip10000 for ASR research, we created manual transcripts of a subset of data: 670 videos of dev set (20 hours) and 566 videos of test set (15 hours). This amount of manually transcribed data can be used to study ASR behaviour on a much wider range of data than is typically the case in ASR research. Videos for manual transcripts were selected from shorter ones available in the corpus to increase the number of documents and the diversity of content in the ASR evaluation set. The selected videos were manually transcribed by crowd-sourcing using Amazon Mechanical Turk (AMT).

6.3.2 Model architecture

This section presents the baseline model architecture used for experiments. The baseline ASR system was a hybrid DNN-HMM system using Kaldi speech recognition software (Povey et al., 2011). This is a standard architecture that was introduced in Section 2.1.2 and Section 2.1.3. Following the ASR architecture, the section describes segmentation applied to raw unsegmented user-generated videos of Blip10000 used for experiments. Finally, the section shows training details of semi-supervised methods to build multi-domain ASR.

Hybrid DNN-HMM systems

Two hybrid DNN-HMM ASR systems were built for our experiments using Kaldi (Povey et al., 2011). The first system was trained on roughly 500 hours of training data of How2 (Sanabria et al., 2018). The second was trained on 960 hours of LibriSpeech audio (Panayotov et al., 2015). Acoustic conditions of How2 data is more varied and similar to that of Blip10000 than LibriSpeech, since Blip10000 consists of user-generated spoken videos. Nevertheless, the domain of How2 is limited to instruction, whereas Blip10000 contains 26 different genres of user-generated spoken content. The acoustic model consisted of 17 time-delay layers with 1,024 hidden units each and trained using LF-MMI (Povey et al., 2016). Acoustic features used in the experiments were standard 40 dimensional MFCCs.

The N-gram language model for the hybrid ASR system was trained on a combination of How2 and LibriSpeech transcripts. This N-gram model was used in the hybrid systems. The value of n was set to 3-gram using the SRILM toolkit (Stolcke, 2002). In addition to the N-gram language model, an RNN language model was trained for lattice re-scoring (Xu et al., 2018). The RNN language model consisted of two LSTM layers with 256 hidden units each and trained on both How2 and LibriSpeech transcripts.

Segmentation details

For semi-supervised training, the untranscribed Blip10000 data needed segmentation for efficient processing of data. The VAD system used for the experiment was the NeMo toolkit (Majumdar and Ginsburg, 2020) trained on the Google Speech Commands and Freesound datasets. This tool is claimed to classify speech and non-speech regions with 99% accuracy[2]. Untranscribed data was split into segments when non-speech frames were longer than 2 seconds and non-speech frames were included in neighbour segments when less than 2 seconds. For each detected speech region, 0.5 of non-speech frames were maintained at the beginning and end of each segment to avoid abrupt cut-offs. Equal segments were created by segmenting untranscribed

[2] https://ngc.nvidia.com/catalog/models/nvidia:vad_matchboxnet_3x1x1

data into 30 second chunks with 5 second overlap with adjacent segments. Segments of 30 seconds were quick to process with the seed system, but 5 seconds of overlap ensures no abrupt cut-offs.

Semi-supervised training

For semi-supervised training, the semi-supervised ASR system was trained on roughly 500 hours of manually transcribed How2 data combined with untranscribed Blip10000 dev data. For each segment created from untranscribed Blip10000 dev data, confidence scores were computed using the method from (Yu et al., 2010) (Section 6.1), and when speech segments below a certain threshold were rejected. The optimal threshold was determined empirically and results are provided in Section 6.4. To determine the optimal confidence level to remove noisy segments of untranscribed speech, 270 hours of Blip10000 dev data randomly selected were combined with How2 training data to run model training and tuning of an optimal confidence score faster. For the rest of semi-supervised experiments, 1,050 hours of Blip10000 dev was combined with How2 training.

The system was trained on the decoded lattices of the Blip10000 data using semi-supervised LF-MMI (Manohar et al., 2018) (Section 6.1), while standard LF-MMI training was applied when using How2 data. The semi-supervised LF-MMI training exploits multiple hypotheses present in a decoder lattice obtained from decoding untranscribed Blip10000 using a seed system. This approach has been demonstrated to be better than training an acoustic model using 1-best hypotheses (Manohar et al., 2018).

6.3.3 Content genre adaptation

Content genre adaptation is hypothesised to train an acoustic model which is more robust to certain acoustic noise observed more frequently in a certain genre. The videos of the Blip10000 corpus are accompanied by a genre tag created by a content creator. For content genre adaptation of the acoustic model, a genre code was generated by transforming the genre tag of each Blip10000 video into a unique digit (e.g., 1: “technology”, 2: “documentary”). Since How2 videos were not classified into different genres and all were instruction videos, How2 speech segments shared the same genre code (i.e., 0). The genre embedding extractor was trained on segments of 1,050 hours of untranscribed Blip10000 each of which was associated with a genre tag and segments of 500 hours of How2 with a uniform genre code. The embedding extractor was trained to classify a segment into 27 genre tags (26 Blip10000 genre tags and 1 How2 tag). The extractor was trained to produce embeddings with the size of 512. This value was used in the original paper on x-vector (Snyder et al., 2018). Also following the x-vector paper, a genre embedding vector was the output of the first segment layer of the extractor. At recognition time, a genre embedding vector is extracted from a speech segment using the extractor and the embedding vector is concatenated with an acoustic feature vector for input of the acoustic model.

6.4 Experimental Results

This section provides experimental results of semi-supervised training and acoustic model adaptation using genre tags. The first sub-section shows WER results of our two hybrid ASR systems trained on How2 and LibriSpeech data. The best system in this sub-section identified in the experiment is used as the seed system to train a semi-supervised system. The second sub-section examines the effects of using different confidence measure to remove noisy speech segments when using semi-supervised training. The third sub-section summarises WER results using the RNN language model to re-score ecoder lattices for semi-supervised acoustic model training. The fourth sub-section provides WER results of semi-supervised training for N-gram and RNN language models The final sub-section presents the semi-supervised ASR system augmented with acoustic model adaptation using user-uploaded genre tags. Unless specified, changes in evaluation metrics reported here

Table 6.1: WERs of the baseline hybrid DNN-HMM systems on transcribed Blip10000 dev and test set.

	blip dev	blip test
hybrid How2	31.27	44.69
hybrid Libri	35.94	51.42

are absolute changes.

6.4.1 Baseline results

As mentioned in Section 6.3.2, two hybrid HMM-DNN systems were built as a candidate of a seed system used to generate decoder lattices for semi-supervised training. One system was trained on 500 hours of How2 data and the second system was trained on 960 hours of LibriSpeech data. Table 6.4.1 presents WER results of the hybrid systems trained on How2 and LibriSpeech. The results show that an acoustic model trained on How2 is more suitable for transcription of Blip10000 data than LibriSpeech data. The How2 system is better than the LibriSpeech system for more than 4% and 6% WERs on Blip dev set and Blip test set, respectively. This can be explained by the fact that both How2 and Blip10000 data are user-generated, while LibriSpeech is an audio-book corpus. Both systems produced higher WERs on the Blip test set. The only condition to select Blip10000 videos for manual transcription was the length of videos. The results indicate that videos chosen for the Blip test set are more challenging to recognise than the Blip dev set. The seed system used for the remainder of the experiments is the How2 hybrid system, since this system produced the best WERs on the transcribed Blip10000 dataset.

6.4.2 Segmentation and confidence score

The Blip10000 videos consist of a mixture of speech utterances and other audio activities. To use untranscribed Blip10000 data for semi-supervised training, it is desirable to segment video data into a fixed length segments and reject segments that are likely to be too noisy for training. For the segmentation approach, we

Table 6.2: WER results of different confidence scores. Utterances with confidence score below {70, 80, 90}% removed. Untrancribed data were segmented using equal segmentation "eq" or VAD segmentation "vad".

	blip dev		blip test	
conf	eq	vad	eq	vad
70	30.69	30.87	44.07	44.22
80	**30.49**	**30.47**	**43.78**	**43.91**
90	30.98	30.86	44.40	44.37

compare segmentation using VAD and equal segmentation. A confidence score of a speech segment is computed by averaging the posterior probabilities of phones from a seed system (Yu et al., 2010) (Section 6.1). In this experiment, we examine three thresholds of a confidence score {70, 80, 90} and reject segments below the threshold.

Table 6.2 shows the effects of different confidence scores to remove unreliable speech segments of untranscribed data. The randomly selected subset of Blip10000 data (270 hours) combined with How2 data (500 hours) was used to train an acoustic model faster (there were 12 combinations of confidence threshold and two segmentation approaches). The results show that setting the confidence score to 80% was the most effective level of removal of noisy speech utterances. Surprisingly, there was not a big difference between equal segmentation and VAD segmentation, although speech segments from created VAD were expected to be cleaner than equal segmentation. The reason for this result is examined in Section 6.5. For the reminder of the experiments, semi-supervised acoustic models were trained on segments created from equal segmentation, from which ones with confidence score below 80% were removed.

6.4.3 RNN language model re-scoring for LF-MMI

As mentioned in Section 6.1, the LF-MMI method exploits decoder lattices of untranscribed data rather than 1-best transcripts for semi-supervised training of an acoustic model (Manohar et al., 2018). Training an acoustic model on decode output means that some of the target phone labels can contain errors. Rather than using

Table 6.3: WER results of semi-supervised training on re-scored decoder lattices using an RNN language model.

	blip dev	blip test
AM-semisup	29.85	42.65
AM-RNN	29.54	42.39

a 1-best hypothesis for an training example, using the whole decoder lattice can mitigate transcription errors as the lattice contains alternative hypotheses. This motivates the use of an RNN language model to improve hypotheses of decoder lattices (Xu et al., 2018), so that a semi-supervised acoustic model can learn from enhanced re-scored decoder lattices.

This experiment examines the use of an RNN language model to improve decoder lattices that are used for target labels of semi-supervised training. For this experiment, an RNN language model was trained on both How2 and LibriSpeech transcripts (Section 6.3.2). For this and the rest of experiments, a semi-supervised acoustic model was trained on 1,050 untranscribed Blip10000 dev set combined with 500 hours of How2 data. The approach to re-scoring decoder lattices for semi-supervised acoustic model training is referred to as "AM-RNN".

Table 6.3 summarises the WER results of RNN re-scoring of decoder lattices for semi-supervised acoustic model training. Re-scoring lattices for semi-supervised training improved the WER of Blip10000 dev set by 0.31% and test set by 0.26%. This approach was shown to bring marginal gain in WERs. For the rest of the sections, "AM-RNN" system is used for following experiments.

6.4.4 Semi-supervised training of N-gram and RNN language model.

Section 6.3.2 discussed semi-supervised training of language models whereby 1-best transcripts of untranscribed data are added to a training corpus of language models (Çelebi and Saraçlar, 2013). This section examines the effects of semi-superivsed training of an N-gram language model and an RNN language model. The procedures

Table 6.4: WER results of semi-supervised training for N-gram and RNN language models.

	blip dev	blip test
ngram	29.54	42.39
RNN	27.70	40.44
semisup-ngram	28.99	41.61
semisup-RNN	**27.28**	**39.68**

to train semi-supervised N-gram and RNN language models are as follows. First, the seed system is used to decode untranscribed Blip10000 data and obtain 1-best transcripts from decoder lattices. Then, these 1-best transcripts are combined with manual transcripts of How2 and LibriSpeech to create a train dataset. While the enhanced N-gram is used as part of a decoding graph (Section 2.1.4), the RNN language model re-scores decoder lattices of the Blip10000 dev and test sets that have manual transcripts for WER evaluation. While an RNN language model in Section 6.4.3 was used to improve decoder lattices for acoustic model training, the RNN language model in this section is used to improve the final WER results. For clarification, an N-gram and an RNN language model without semi-supervised training is referred to as “ngram” and “RNN”, while with semi-supervised “semisup-ngram” and “semisup-RNN”.

Table 6.4 summarises WER results of semi-supervised training of language models. The “AM-RNN” using re-scored decoder lattices for acoustic model training in Table 6.3 corresponds to “ngram” in Table 6.4. Using semi-supervised training for an N-gram language model “semisup-ngram” led to 0.55% WER reduction for the dev set and 0.78% for the test set over a vanilla N-gram model “ngram”. The largest WER improvement of 1.5-2.0% WER reduction was obtained by RNN language model re-scoring which was also observed in the work from Xu et al. (2018). Semi-supervised training for an RNN language model “semisup-RNN” brought 0.42% WER improvement on blip dev and 0.76% over a vanilla RNN language model “RNN”. Overall, these methods together improved WERs by 2.26% on blip dev and 2.71% on blip test.

Table 6.5: WER results of acoustic model adaptation using a genre code and a genre embedding vector.

	blip dev	blip test
baseline	27.28	39.68
genre-code	27.00	39.46
genre-emb	**26.82**	**39.21**

6.4.5 Content genre adaptation

As discussed in Section 6.3.3, content genres are hypothesised to provide a neural acoustic model with information about acoustic noise that is more likely to appear in a certain content genre (e.g., "applauds" in "conference"). We propose two adaptation techniques using content genres: genre code and genre embedding. A genre code is appended as a unique genre code to an acoustic feature vector. A genre embedding approach is to extract an embedding vector using a genre classification network described in Section 6.3.3. This experiment is incremental to all of the previous experiments, meaning that the "baseline" system was trained on 1,050 hours of untranscribed Blip10000 dev combined with 500 hours of How2 and semi-supervised N-gram and RNN language models were used for decoding and lattice re-scoring. The "baseline" system augmented using genre code is referred to as "genre-code" and augmented using genre embedding is referred to as "genre-emb".

Table 6.5 summarises WER results of acoustic model adaptation using genre code and genre embedding. Using the genre code brought WER improvement of 0.28% on blip dev and 0.22% on blip test. The genre embedding feature brought a large WER improvement than the genre code; gain in WER 0.46% on blip test and 0.47% on blip test over the "baseline" system. The experimental results demonstrate that both the genre code and the genre embedding feature could bring marginal improvement to an acoustic model.

6.5 Discussion and Conclusions

SCR systems often operate on broad topics of content and it is desirable to develop ASR that can accurately transcribe broad domains of content for SCR effectiveness. For such broad topics of spoken content, it is challenging to have high coverage of manual transcripts for ASR training which is costly and time-consuming to create. The goal of our investigation in this chapter was to achieve multi-domain ASR using semi-supervised training and acoustic model adaptation using content genre information.

Our investigation thoroughly examined the flowchart of semi-supervised training for acoustic and language models. Natural audio data are not segmented into speech utterances and regions of speech are unknown in spoken content. For semi-supervised training, it is also desirable to reject noisy speech segments which can potentially hinder efficiency of acoustic model training. We compared two segmentation approaches: segment raw audio into equal segments and VAD-based segmentation. The results in Section 6.4.2 show small differences between the two segmentation approaches. For rejection of potentially noisy segments, rejecting segments below confidence score 80% was found to be effective.

The optimal configuration of semi-supervised training and genre-based acoustic model adaptation found in the experiments are as follows.

- Semi-supervised training using 1,050 hours of untranscribed Blip10000 data; WER improvement 1.42% on blip dev and 2.04% on blip test
- Re-scoring decoder lattices for semi-supervised acoustic model training; WER improvement 0.31% on blip dev and 0.26% on blip test
- Semi-supervised N-gram; WER improvement 0.55% on blip dev and 0.78% on blip test
- Semi-supervised RNN language model; WER improvement 0.42% on blip dev and 0.76% on blip test

- Genre embedding adaptation; WER improvement 0.46% on blip dev and 0.47% on blip test

Overall, WER improvement from the hybrid system trained on How2 data was from WER 31.27% to 26.82% on blip dev and from WER 44.69% to 39.21% on blip test. It is known that WER higher than 30% can often hinder search effectiveness (Larson and Jones, 2012) and a linear relationship between WERs and MAP scores was found in TREC SDR Tracks (Garofolo et al., 2000). WER improvement obtained by semi-supervised training and genre-based adaptation is evaluated in the context of SCR in Chapter 7.

Chapter 7

Augmentation of SCR ranking methods

SCR is a task to satisfy user information needs whereby a ranking model is used to retrieve spoken content in which users are potentially interested. The unique challenge posed for the task is presence of ASR transcription errors contained in transcripts of spoken documents. Chapter 5 and Chapter 6 explored incorporation of multimodal information into ASR, semi-supervised training and genre-based acoustic model adaptation for improvement of ASR transcripts with the goal of SCR effectiveness. This chapter examines different types of transcripts of speech documents and different ranking models for SCR effectiveness.

This chapter presents two research investigations. Firstly, the chapter presents an investigation into the use of manual transcripts, standard ASR transcripts and semi-supervised ASR transcripts for the standard BM25 model in the SCR task. The goal of this investigation is to examine how SCR effectiveness is affected by the level of noise in each transcript. Secondly, the chapter presents a research investigation into the use of two state-of-the-art neural ranking models, DRMM and PACRR and transformer-based ranking models in the SCR task. Both the neural ranking models and transformer-based ranking models have been demonstrated to outperform traditional probabilistic models such as BM25 (Guo et al., 2020; Lin et al., 2020). Although transformer-based ranking models have been applied to the podcast retrieval task in TREC 2020 and 2021 (Jones et al., 2020, 2021), due to the absence of manual transcripts in the task, it is not yet clear if those advanced models could overcome ASR transcription errors. Further, we propose an extension of those models using ASR N-best transcripts to mitigate the effects of transcription errors contained in ASR transcripts. Our initial work on neural ranking models for SCR was accepted for publications on the 2021 Automatic Speech Recognition and Understanding (ASRU) conference[1].

The reminder of this chapter is organised as follows. The first section introduces an extension of neural ranking models and transformer-based ranking models using ASR N-best hypotheses. Our motivation for using ASR N-best hypotheses for search instead of 1-best is that lower order hypotheses of ASR transcripts may contain information missing from the 1-best transcripts. The second section presents the details of corpora used for SCR experiments. Our experiments are carried out on the How2 corpus (Sanabria et al., 2018) and the Blip10000 corpus (Schmiedeke et al., 2013). Although these corpora were used in the ASR experiments in Chapter 5 and Chapter 6, this section shows the details of the corpora relating to SCR experiments. The third section summarises the results of SCR experiments with the BM25 model using manual transcripts, standard ASR transcripts and semi-supervised ASR transcripts. The goal of this experiment is to examine the gap in search effectiveness between these types of transcripts. The final section of this chapter presents experimental results of neural ranking models and transformer-based ranking models on the How2 and Blip10000 data with and without an N-best extension.

[1] My contribution is background research, designing and conducting experiments and analysis of the experimental results.

7.1 Extension of Ranking Models using ASR N-best Hypotheses

Chapter 3 introduced recently proposed advanced ranking models including neural ranking models (Guo et al., 2020) and transformer-based ranking models (Lin et al., 2020). In the literature, these models have been demonstrated to outperform the traditional probabilistic models such as the BM25 model (Guo et al., 2020; Lin et al., 2020). The use of transformer-based ranking models has been seen in the recent TREC 2020-2021 Podcasts Tracks (Jones et al., 2020, 2021). However, ASR transcripts of podcasts data provided in the tasks were limited to 1-best hypotheses. This section considers an extension of neural ranking models and transformer-based ranking models using ASR N-best transcripts to mitigate the effects of errors contained in ASR hypotheses.

This section first introduces N-best extension of DRMM followed by N-best extension of PACRR model. The goal of these extensions is to enable the two neural ranking models to take as input N-best hypotheses instead of 1-best hypotheses. This section then presents an extension of transformer-based ranking models. As shown in Section 3.3.2, transformer-based ranking models can be classified into two: re-ranking system and dense retrieval (DR) system. Although our N-best extension could be used for both the transformer-based ranking models, this section proposes the extension only for the DR system, since we found in preliminary experiments that the N-best extension of the re-ranking system was empirically not as effective as the DR system.

7.1.1 N-best extension of DRMM

DRMM is a neural ranking model that transforms a query-document similarity matrix into a matching histogram as input of a feed-forward network for relevance score computation (Guo et al., 2016). The standard DRMM model takes a single document D as input along with a query Q to produce a relevance score. In our

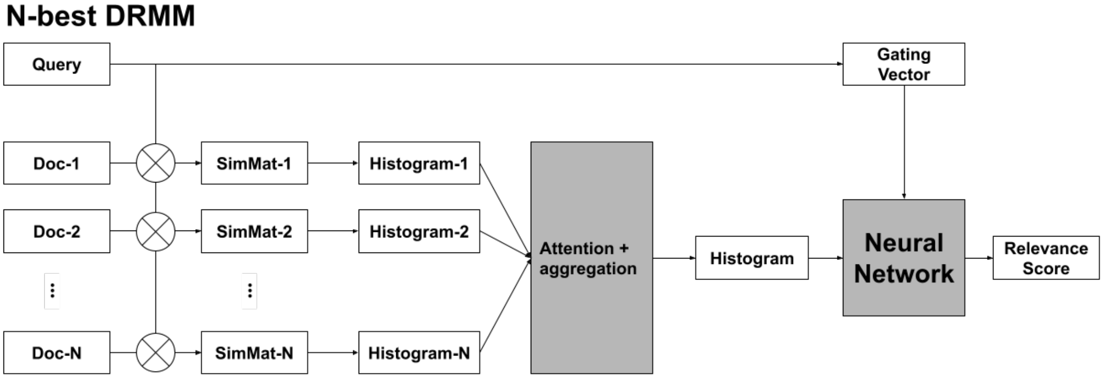


Figure 7.1: N-best version of DRMM. Grey squares indicate these are learnable parameters of the model. The crossing of Query and Doc is interaction of query terms and document terms. SimMat is a similarity matrix of query terms and document terms.

N-best configuration, N transcripts for each spoken document D are used. The core idea of the N-best DRMM is to apply attention weights to N matching histograms, and to aggregate them to form a single matching histogram as an input of linear layers. This enables the ranking model to produce a relevance score which takes account of all N hypotheses of document D. The output of the linear layers is a relevance score from DRMM.

Figure 7.1 shows the N-best extension of DRMM. In the N-best settings, each of the N-best transcripts of D is transformed into N similarity matrices S_1, S_2 , ..., S_N. As mentioned in Section 3.2.2, the input of a DRMM is a fixed length matching histogram. Therefore, each row of the similarity matrices is transformed into a histogram of b bins. This produces N matching histograms $M_1, M_2, ..., M_N$ with size $M_i^{|Q|\times|b|}$. The goal here is to apply an attention mechanism, aggregate N matching histograms and form a single matching histogram M_D in which information from N documents is gathered. This can be expressed as follows,

$$u_{ij} = \mathbf{W_n}\mathbf{m}_{ij} + \mathbf{b} \tag{7.1}$$

$$a_{ij} = Softmax(u_{ij}) \tag{7.2}$$

$$\mathbf{m}_{Dj} = \sum_{i=1}^{N} a_{ij}\mathbf{m}_{ij} \tag{7.3}$$

where $\mathbf{m_{ij}}$ is a vector of the ith document of the jth query's matching histogram (jth row of M_i), $\mathbf{W_n}$ is a weight matrix, $\mathbf{b}$ is a bias term and a_{ij} is a weight vector for $\mathbf{m}_{ij}$. The aggregated matching histogram of N-best documents $\mathbf{M}_D$ is input to a neural network. The rest of process to produce a relevance score is identical to Equation 3.4-Equation 3.6.

7.1.2 N-best extension of PACRR

The PACRR model directly takes a similarity matrix as input to produce a query-document representation. This representation is input of a feed-forward network to compute a relevance score (Hui et al., 2017). In N-best settings, the convolutional layer directly takes N similarity matrices to produce a query-document representation. Figure 7.2 shows our N-best extension of PACRR. Similar to N-best DRMM, each of the N-best transcripts is transformed into a similarity matrix $S_1, S_2, ..., S_N$. A convolutional layer of PACRR can directly take N similarity matrices as channels. This is analogous to image recognition where an input image typically consists of red, green, blue (RGB) channels; hence three channels. Instead of three, the number of input channels of convolutional layers for N-best PACRR is set to N. This input is formed by stacking N similarity matrices $S_1, ..., S_N$ and creating a 3D tensor S. The output of the convolutional operations is input of two max pooling layers. This is the same as PACRR for 1-best transcript in Section 3.2.3, except that N unigram similarity matrices are flattened and l_s salient values are retained.

7.1.3 N-best extension of BERT DR systems

Ranking models using a transformer architecture have been shown to be more effective for IR tasks than traditional probabilistic models and neural ranking models (Lin et al., 2020). Two types of ranking models using a transformer model, specifically the Bidirectional Encoder Representation from Transformer (BERT), are reranking (Nogueira and Cho, 2019) and DR (Karpukhin et al., 2020) systems. Similar

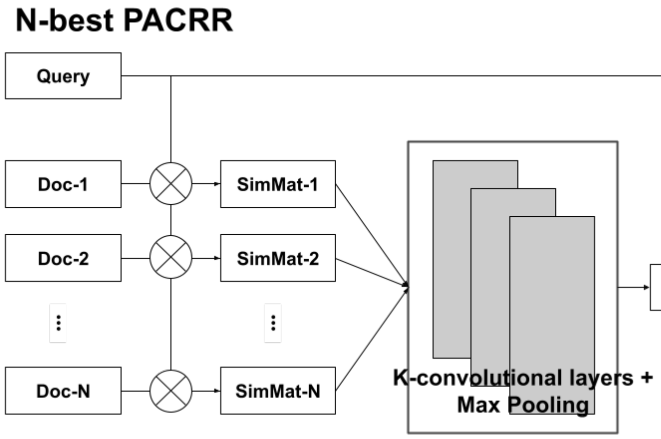


Figure 7.2: N-best version of PACRR. Grey squares indicate these are learnable paramters of the model. The crossing of Query and Doc is interaction of query terms and document terms. SimMat is a similarity matrix of query terms and document terms.

to neural ranking models, an ASR N-best extension can be considered for BERT-based ranking models. This section proposes an N-best extension for the BERT DR system. Two approaches to fusing N-best transcripts are early fusion and late fusion. This section introduces the N-best extension only to the DR system, since the BERT re-ranking system empirically did not benefit from early or late fusion of N-best transcripts in our exploratory experiments.

Existing work has explored early fusion of ASR N-best transcripts for Spoken Language Understanding (Ganesan et al., 2021) and for ASR error correction (Zhu et al., 2021). In Spoken Language Understanding, an input sequence for BERT is often short and N-best ASR hypotheses can be concatenated with the $[SEP]$ token to form a single input (Ganesan et al., 2021). However, each ASR hypothesis for the SCR task is often long and this approach cannot be applied to the DR BERT system. Our early fusion approach employs alignment of ASR N-best hypotheses from Zhu et al. (2021). This approach changes the length of input sequence only with a few tokens and the DR BERT model can still take it as input without dropping a significant number of tokens. The reminder of this section describes early fusion and late fusion of ASR N-best hypotheses for the BERT DR system.

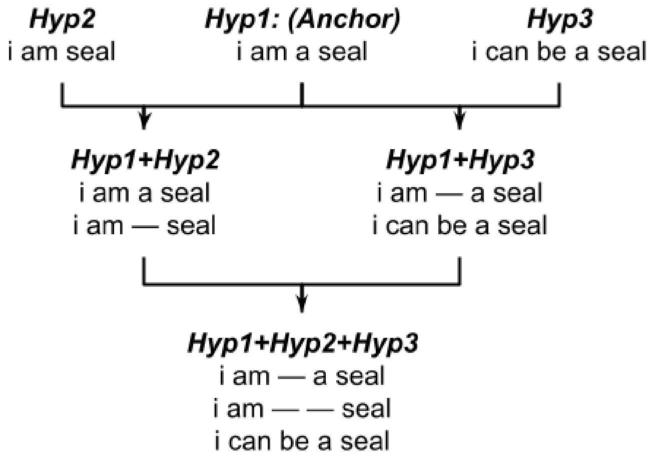


Figure 7.3: Diagram illustrating alignment of ASR N-best transcripts to form an input sequence of the DR BERT model .

Early fusion

The early fusion of N-best transcripts for the BERT DR model uses alignment of ASR N-best transcripts. This is due to the constraints on the input length of the BERT model (Section 3.3.1). Document input for the DR BERT model often consists of a large number of tokens and concatenating ASR N-best transcripts can truncate a large amount of tokens due to the length limit.

Figure 7.3 illustrates an example alignment approach to fusing 3-best ASR hypotheses. In the figure, the 1-best transcript is used as an anchor transcript and aligned with the other two transcripts. The aligned transcripts are further combined to match the length across N-best transcripts. As with Zhu et al. (2021), "—" is used as a special token to alleviate length mismatch between transcripts. Once the N-best transcripts have been aligned, the tokens are transformed into word embeddings and the ith position of aligned tokens can be combined by taking the average. For example, in Figure 7.3, the first position of the three transcripts is all "i" and this results in the averaged embeddings of "i".

Late fusion

Our late fusion approach to using N-best transcripts for the BERT DR model creates N indexes for ith ranked ASR hypotheses. Figure 7.4 illustrates the late fusion

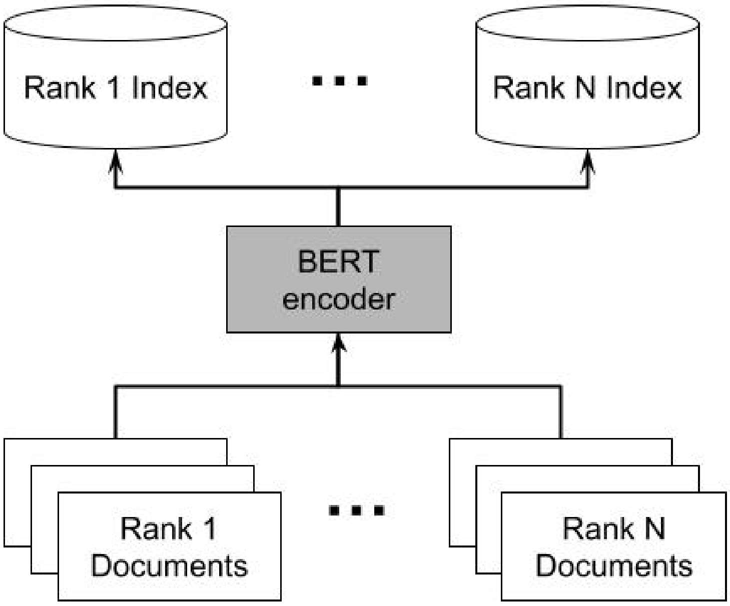


Figure 7.4: Diagram illustrating late fusion approach to integration of ASR N-best transcripts into the BERT DR model.

approach to the extension of the BERT DR model using ASR N-best transcripts. The BERT encoder is used to encode ith rank documents to create the ith index. For example, rank-1 transcripts are all encoded to create a rank-1 index in Figure 7.4. At search time, the BERT model encodes a query into a vector and the ANN model runs search for all N indexes. The final result is computed by summing the scores for each document from each index and ranking the summed scores in descending order.

7.2 Details of SCR Test Collections

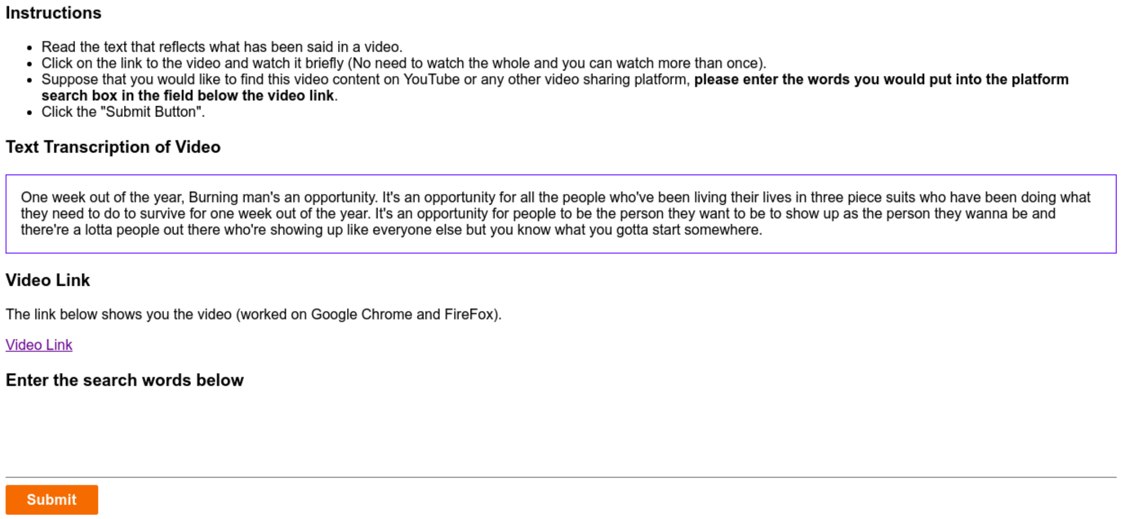
This section provides details of our datasets used in the SCR experiments. The How2 corpus and the Blip10000 corpus were used for the ASR experiments in Chapter 5 and Chapter 6. This section describes details of these corpora relating to SCR studies including training queries used for ranking model training and evaluation of ranking models.

7.2.1 How2

The How2 corpus consists of 19,770 instruction videos, manual transcripts and their titles[2]. In our How2 experiments, the 19,770 video titles were used as queries. For training and evaluation of neural ranking models and transformer ranking models, a set of randomly selected 500 titles were reserved as test queries and the rest of the titles were used for model training. The BM25 model does not require model training and when evaluating output of the BM25 model, the same 500 titles were used as evaluation queries.

Three types of How2 transcripts were used for the SCR experiments: manual transcripts, standard ASR transcripts and semi-supervised ASR transcripts. The standard ASR transcripts of How2 were created using the hybrid HMM-DNN ASR system trained on 960 hours of LibriSpeech data (Panayotov et al., 2015). This is the scenario where the ASR system available was not trained on the in-domain data of How2, since LibriSpeech is an audio-book corpus. The WER of the standard ASR transcripts was 31.23% on How2 dev and 30.86% on How2 test. The semi-supervised ASR transcripts were created using a combination of 960 hours of LibriSpeech data and 500 hours of untranscribed How2 data and the LibriSpeech system as a seed system. The methods to train a semi-supervised ASR system are as described in Chapter 6, except that content genre-based adaptation was not used because the genre of How2 data was all instructions. The WER of semi-supervised ASR transcripts was 24.13% on How2 dev and 23.77% on How2 test. For an extension of neural ranking and transformer ranking models using ASR N-best hypotheses, {5, 10, 20}-best transcripts were generated using the above standard ASR system and semi-supervised ASR system.

[2] Some of the videos may no longer be available on YouTube.



Instructions

- Read the text that reflects what has been said in a video.
- Click on the link to the video and watch it briefly (No need to watch the whole and you can watch more than once).
- Suppose that you would like to find this video content on YouTube or any other video sharing platform, **please enter the words you would put into the platform search box in the field below the video link**.
- Click the "Submit Button".

Text Transcription of Video

One week out of the year, Burning man's an opportunity. It's an opportunity for all the people who've been living their lives in three piece suits who have been doing what they need to do to survive for one week out of the year. It's an opportunity for people to be the person they want to be to show up as the person they wanna be and there're a lotta people out there who're showing up like everyone else but you know what you gotta start somewhere.

Video Link

The link below shows you the video (worked on Google Chrome and FireFox).

Video Link

Enter the search words below

Figure 7.5: An example screenshot of platform to collect queries from Amazon Mechanical Turk Workers. Workers were presented with a video transcript and a link to the video to enter search query terms.

7.2.2 Blip10000

The Blip10000 corpus (Schmiedeke et al., 2013) was used for evaluation of neural ranking models and transformer ranking models trained on How2 data. The goal of this experiment was to evaluate the ranking models in a domain-mismatch scenario for a known item search task. The WER of the standard ASR transcripts was 31.27% on blip dev and 44.69% on blip test. The WER of the semi-supervised ASR transcripts using content genre adaptation in Section 6.4.5 was 26.82% on blip dev subset and 39.21% on blip test subset. Manually created transcripts are not available for the whole corpus. Although the number of available videos in the Blip10000 corpus was 14,838, the total number of standard ASR and semi-supervised ASR transcripts as a Blip10000 collection was 6,432 documents, because videos could contain a foreign language, corrupted audio or speech with loud noise in background. In these cases, the ASR decode process could fail to create transcripts.

For search experiments, 15 known item queries for dev and 35 queries for test were created through Amazon Mechanical Turk (AMT). These 50 documents were selected randomly from transcribed parts of the Blip10000 dataset (Section 6.3.1). Figure 7.5 shows an example platform for query collection from AMT Workers. AMT

Table 7.1: MRR results of the How2 and Blip10000 known-item search task using the BM25 model with comparison of manual, standard ASR and semi-supervised ASR transcripts.

transcript	How2		Blip10000	
	WER	MRR	WER	MRR
manual	0.0	40.27	–	–
semi-supervised ASR	23.95	30.85	32.75	29.22
standard ASR	31.05	29.37	37.98	17.15

Workers were asked to enter at least 3 query terms suppose they wish to search for a video presented in the screen. AMT Workers were also asked not to participate in query term creation more than 4 times to gather a diverse set of query terms.

7.3 Investigation using Standard BM25 IR model

This section presents the first investigation into comparison of the three types of transcripts with respect to SCR effectiveness using the BM25 model (Robertson et al., 1995). The three types of transcripts for comparison are manual transcripts, standard ASR transcripts and semi-supervised ASR transcripts. The standard BM25 model is applied to the known-item search task on How2 and Blip10000. The BM25 model for the experiment is implemented using the python binding of Lucene library. The $k1$ and b values for the BM25 model (Section 3.1) were constant 1.2 and 0.75, which are the default parameters of the Lucene library. The search results are reported using the MRR metrics suitable for the known-item search task (Section 3.4.1). Unless specified, changes in evaluation metrics reported here are absolute changes.

7.3.1 Experimental results

The How2 known-item search uses 500 video titles as queries. These 500 titles were randomly selected and reserved as evaluation queries. These evaluation queries are also used for the experiment with neural ranking and transformer ranking models. The Blip10000 known-item search uses 50 known-item queries.

Table 7.1 summarises WERs and MRR scores of the known-item search task on How2 and Blip10000. The WERs reported in the Table are the average WER of dev and test set on How2 and Blip10000. Manual transcripts are not available for the whole Blip10000 corpus. In How2 experiments, the MRR score difference was by 9.42 between manual and semi-supervised ASR transcripts. Although there was around WER 7% difference between semi-supervised ASR transcripts and standard ASR transcripts on How2, improvement of the MRR score was just over 1%. On the other hand, improvement of the MRR score in the Blip experiment was large; around 5% improvement of the WER led to roughly 12% increase in the MRR score. These results indicate that semi-supervised transcripts created using the techniques described in Chapter 6 improve search effectiveness.

7.4 Investigation using Neural Ranking and Transformer Models

This section presents our second investigation into the use of neural ranking and transformer ranking models for the SCR tasks. As mentioned in Chapter 3, neural ranking models and BERT ranking models are promising alternative ranking models that bring more search effectiveness than the traditional probabilistic models including the BM25 model (Guo et al., 2020; Lin et al., 2020). The goal of this research investigation is (i) to examine the effects of ASR errors contained in transcripts on neural ranking models and BERT ranking models and (ii) to measure SCR effectiveness of our proposed ASR N-best extension to neural ranking models and BERT ranking models. It is hypothesised that using N-best transcripts instead of 1-best enable these models to be robust to transcription errors. This first sub-section describes architecture of the models, followed by training details of the models. The second sub-section presents 1-best experiments with the models on How2 data. The third sub-section summarises N-best experiments with the models on How2 data. The last sub-section presents SCR experiments using the neural and transformer ranking models on Blip10000 data as a domain-mismatch scenario.

7.4.1 Model architecture

This section provides details of the architecture of our neural ranking models and transformer ranking models. Neural ranking models used in this investigation are DRMM and PACRR (Guo et al., 2016; Hui et al., 2017). This section describes the hyper-parameters for both models and configuration of the gating vector incorporated into the neural ranking models. Secondly, the section provides hyper-parameter details of two transformer ranking models, BERT re-ranking and BERT DR models.

DRMM architecture

As shown in Section 3.2.2, DRMM requires the size of a matching histogram and layer sizes of a feed-forward network as hyper-parameters. In our experiment, the feed-forward network of DRMM had an input layer size of 5, a middle layer size of 30 followed by two hidden layers of size 5 and 1, following the architecture of the original paper (Guo et al., 2016). Although the original paper from Guo et al. (2016) used the size of a matching histogram 20, this size was set to 5 in our experiment, as it was empirically in our preliminary studies to be a better value than 20.

PACRR architecture

The PACRR model requires the following parameters to determine its model architecture (Section 3.2.3): (i) the maximum number of document terms l_d, (ii) the number of filters of the convolutional layers l_f, (iii) the kernel size of convolutional layers l_g and (iv) the number of salient values to be kept at second max pooling l_s. Unless specified, these hyper-parameters were empirically chosen by hyper-parameter tuning of the PACRR model in the How2 experimental settings. In the original paper of Hui et al. (2017), the maximum number of document terms l_d was fixed at 768. For PACRR experiments, a value close to the original paper, 700, was chosen. In initial experiments, the best values of l_f and l_g were 16 and 3. The l_g value 3 means that bi-gram convolution and tri-gram convolution were applied to the input matching matrix. Regarding the parameter l_s, our preliminary experiments on How2 indicated that the value 3 was the best in 1-best settings, while the value 7 was the best in N-best settings. This indicates that N-best documents contained more useful information than 1-best documents and keeping more salient values by l_s benefited search effectiveness of PACRR.

Gating vector

As described in Section 3.2.2 and in Section 3.2.3, the gating vector for DRMM and PACRR is a weighting vector to determine importance of each query term. The gating vector consists of a word embedding vector of each query term concatenated with its IDF value. An attention layer (Equation 3.3) is then applied to the gating vector to compute weights of query terms. Our experiment employed the pre-trained Glove model with each word embedding vector of 300 dimension (Pennington et al., 2014). IDF values were computed from manual transcripts or standard ASR transcripts of the How2 corpus. The use of ASR transcripts for IDF computation is encouraged to remove the requirement of manual transcripts for model training.

BERT re-ranking and BERT DR

Our transformer re-ranking and DR systems used the BERT model as a pre-trained model. The BERT model, as described in Section 3.3, consists of several transformer blocks and pre-trained on masked word prediction and next sentence prediction tasks (Devlin et al., 2019). BERT is a powerful large language model and can be fine-tuned to various other NLP tasks. We fine-tuned the pre-trained BERT "bert-base-uncased"[3] available in the HuggingFace library (Wolf et al., 2020) for both re-ranking and DR BERT models. This is a widely used BERT model (over 10 million downloads). The "bert-base-uncased" model consists of 12 transformer blocks with each block size 768 and the number of attention heads incorporated in the transformer blocks is 12. This BERT model was pre-trained on masked word

[3] https://huggingface.co/bert-base-uncased

prediction and next sentence prediction using over 11 thousands of English books and on English Wikipedia pages. The length limit of input was 512 tokens and input sequences longer than this limit were truncated. This length limit could be improved as future work.

For the DR system, the BERT model is used to encode queries and documents into vectors and a vector similarity runs to rank document vectors for each query vector (Section 3.3.2). Generally, this secondary search model is Approximate Nearest Neighbour (ANN) (Karpukhin et al., 2020) which runs much faster than BERT re-ranking for inference. We used Faiss software (Johnson et al., 2019) to run ANN similarity search over a query vector and an index of document vectors.

7.4.2 Model training

This section presents training details of the four ranking models: DRMM, PACRR, BERT re-ranking and BERT DR. Unlike the BM25 model, these ranking models require model training on query-document pairs. As described in Section 3.2.4 and Section 3.3.3, DRMM and PACRR models are trained using a pair-wise ranking loss where the models are trained to produce a relevance score 1 for pairs of query and relevant document and a relevance score 0 for pairs of query and irrelevant document. The BERT re-ranking model is trained using a binary cross entropy which is conceptually similar to a pair-wise ranking loss, but a single training example consists of a query-document pair unlike the pair-wise loss. The BERT DR model is trained using a negative log likelihood to maximise similarities of vectorised queries and vectorised relevant documents, while to minimise vectorised queries and vectorised irrelevant documents. The rest of this section provides training details of DRMM and PACRR, BERT re-ranking and BERT DR.

Training DRMM and PACRR

Training hyper-parameters below were found empirically in initial experiments on model tuning, though some parameters were borrowed from the original papers of DRMM (Guo et al., 2016) and PACRR (Hui et al., 2017). Both DRMM and PACRR models were trained on 19,270 pairs of How2 video titles as queries and corresponding documents. As training documents, manual transcripts and standard ASR transcripts of How2 were used to investigate whether these models could be trained on ASR transcripts. DRMM and PACRR models were trained for 30 epochs, as these models were converged within 30 epochs. The initial learning rate 0.001 was found to be effective for DRMM while 0.0005 for PACRR. The Adagrad optimiser was used following Guo et al. (2016). The mini-batch size was 100. For each training query, the top 10 ranked documents selected by the BM25 model (Robertson et al., 1995) were used as negative examples. If a target document for a given input query was included in the list of top 10 documents, the 11th document in a ranked list was taken as 10th negative document.

To monitor performance of neural ranking models, 100 query-document pairs were randomly selected from training query-document pairs as a validation set and the average of the MRR value was computed after each epoch. For each query document pair of the validation set, neural models were used to re-rank the top 500 documents in a ranked list returned by BM25 for MRR computation. The final version of the model was chosen based on the validation MRR score.

Training BERT re-ranking

The BERT re-ranking model was trained using a binary cross entropy as the search task is treated as a binary classification task to decide whether a given query-document pair is relevant or not (Nogueira and Cho, 2019). Similar to DRMM and PACRR training, initial experiments to tune model hyper-parameters were conducted and these parameters described below led empirically to the best model. Out of 19,270 How2 titles, as described previously 500 titles as queries were reserved for model validation. More validation examples were used, since BERT re-ranking was harder to monitor train progress with 100 queries. The re-ranker BERT was trained with 20 epochs and used the model checkpoint which produced the best validation score. Model training was generally converged by epoch 15. The best initial learning rate was empirically selected and it was 3×10^{-7}. For each train query, top 20 documents returned by the BM25 model were selected as negative examples (when a relevant document to train query was included in top20, used 21th document as a negative example). The model could be further improved by using more negative examples, but increasing the number of negative examples exceeded GPU's memory limit and led to slow speed of model training. The AdamW optimizer was used with the weight decay of 5×10^{-5} to avoid overfitting.

Training BERT DR

The BERT DR model was trained using a negative log likelihood to assimilate a vector of relevant document to a vector of query. The DR BERT models were trained for 10 epochs as the models were converged by 10 epochs. The best initial learning rate found for BERT DR was 1×10^{-5}. Similar to BERT re-ranking, the 500 reserved queries were used for model validation, with the number of negative examples used set to 20 and the AdamW optimiser with the weight decay of 5×10^{-5} employed.

For the BERT DR model, we further experimented with Approximate nearest neighbour Negative Contrastive Estimation (ANCE) introduced in Section 3.3.3 (Xiong et al., 2021). The ANCE training updates negative examples at each interval set as a hyper-parameter using the BERT DR model being trained. In our experiment with ANCE training, after each training epoch, all of the negative examples were updated.

7.5 Experimental Results

The SCR experiments with DRMM, PACRR, BERT re-rank and BERT DR models were conducted on the How2 and Blip10000 data similar to Section 7.3. These models were trained on 19,270 pairs of video titles (referred to as "queries" from now on) and transcripts, while 500 How2 video titles were used for model evaluation on How2. The task was the known-item search where each query was associated with one spoken document. The 50 known-item queries of Blip10000 were reserved for model evaluation in the out-of-domain scenario. These models were trained on How2 manual transcripts and How2 standard ASR transcripts to examine the effects of types of transcripts used for search effectiveness. Using ASR transcripts for model training is motivated to avoid preparing manual transcripts for model training that is often expensive and time-consuming. The results in this section demonstrate that ASR transcripts could be used for search model training.

For each evaluation query, the BM25 model was used to obtain the top 1,000 initial ranked documents. DRMM, PACRR and BERT re-ranking models were used to re-rank these 1,000 documents for the final ranked list of documents. The BERT DR model was, on the other hand, used as an encoder of queries and documents (Section 3.3.2). Therefore, each evaluation query was encoded into a vector by the BERT DR model, and the vectorised query was compared against each document vector in an index using approximate nearest neighbour (ANN) search for creation of the ranked list of documents.

Presentation of the experimental results in this section is organised as follows. The first section provides MRR results of How2 known-item search using the neural search models. This section examines the use of ASR transcripts for model training in comparison to the use of manual transcripts. The second section summarises results of the ASR N-best extension. As mentioned in Section 7.1, this method was applied to DRMM, PACRR and BERT DR models and not to BERT re-ranking, since MRR improvement brought to the N-best BERT re-ranking model was empirically very little to no effects. The third section demonstrates search effectiveness of these models on How2 manual, standard ASR and semi-supervised ASR transcripts. The fourth section shows the MRR scores of Blip10000 using the best search models found in the How2 experiments in a domain mismatch scenario.

Table 7.2: MRR results of DRMM, PACRR, BERT-reranking and BERT-DR models on How2 data. Either manual transcripts or ASR transcripts were used for training data of the models and these models were used to rank standard ASR transcripts. The baseline BM25 result of ranking standard ASR transcripts is from Table 7.1

model	MRR train data	
	manual	ASR
BM25	29.37	
DRMM	30.45	30.44
PACRR	32.65	33.62
BERT-rerank	43.03	46.37
BERT-DR	36.25	34.81
BERT-DR-ANCE	47.86	52.41

7.5.1 MRR results of neural ranking and transformer ranking models on How2.

This section summarises MRR results of DRMM, PACRR, BERT re-ranking and BERT DR models on the How2 known-item search task. Training data for these models were either manual transcripts or standard ASR transcripts. When evaluating these models, the transcripts used were standard ASR transcripts to conduct the standard SCR experiments where manual transcripts were not available. Table 7.2 summarises the MRR results of DRMM, PACRR, BERT-rerank and BERT-DR models on How2 evaluation of 500 queries. Two key points from the results are: (i) overall MRR scores of the models and (ii) using ASR transcripts for model training.

(i) The MRR results show that all of the neural and transformer models improved the MRR score over that of the BM25 model. The largest gain is from the BERT-DR model using ANCE training (Xiong et al., 2021). This model improved the MRR score over 20 than that of the BM25. The result from the BRET-DR model using ANCE indicates that selection of negative examples is important for the BERT DR model. The BERT-rerank model produces the second highest MRR score of 46.37. Compared to these models, gain in the MRR score from DRMM, PACRR and BERT-DR without ANCE were small.

(ii) The results demonstrate that the neural ranking and BERT ranking models can be trained on ASR transcripts. All but BERT-DR models produced a higher or almost equal MRR score when the models were trained on standard ASR transcripts. The BERT-DR model produces around 1.5 MRR worse than the model trained on manual transcripts. These results indicate that it is possible to avoid creating expensive manual transcripts for development of neural ranking and BERT ranking models. For the remainder of this investigation, results from the neural models that were trained on ASR transcripts are presented.

7.5.2 MRR results of N-best extension on How2.

This section summarises results for the experiments with the N-best extension of DRMM, PACRR and BERT-DR models. As discussed in Section 7.1, ASR transcripts can still contain transcription errors despite recent improvements in ASR systems. In particular ASR errors can appear more often when there is a mismatch with the domains of training data and evaluation data for ASR systems (Chapter 6.1). The use of N-best transcripts is motivated for the hypothesis that terms missing in top ranked ASR hypotheses can be present in lower ranked ASR hypotheses and using N-best can recover "keywords" for search given a query. The N-best sizes used for model training were {5,10,20}.

Table 7.3 summarises the use of N-best transcripts for DRMM, PACRR and BERT-DR models. Overall, using N-best transcripts generally improved the MRR scores than the 1-best models. The best MRR score observed was from the late fusion of 20-best transcripts for the BERT-DR model trained using ANCE. The MRR score was improved from 52.41 to 55.40. In this method, the BERT model encoded ith ranked documents into an ith index and for each query, N scores were summarised to comptue the final score (Section 7.1). The early fusion method for BERT DR, on the other hand, was not as effective as late fusion models. DRMM and PACRR models also benefited from the N-best extension. the PACRR model in particular improves the MRR score by almost 3 points using 10-best transcripts. However, increasing the N-size from 10 to 20 had a negative impact on DRMM and

Table 7.3: MRR results of DRMM, PACRR and BERT-DR models using the N-best extension on How2 known-item task.

model	MRR N-size			
	1	5	10	20
BM25	29.37			
DRMM	30.44	30.40	**31.86**	31.59
PACRR	33.62	33.88	**36.59**	34.69
BERT-DR-early	34.81	**35.42**	33.71	34.24
BERT-DR-late	34.81	36.94	37.55	**38.20**
BERT-DR-ANCE-early	52.41	52.14	52.75	**54.45**
BERT-DR-ANCE-late	52.41	54.88	54.86	**55.40**

PACRR models. It is likely that the additional lower scoring hypotheses contain too many incorrect hypotheses while reduce search effectiveness.

7.5.3 MRR results of different transcripts on How2.

The last section of How2 experiments presents the effects of using manual transcripts, semi-supervised ASR transcripts and standard ASR transcripts for the neural ranking and BERT ranking models. This experiment is similar to the one conducted using the BM25 model in Section 7.3. The goal of this experiment is to examine how the level of transcription errors in each transcript affects neural ranking and BERT ranking models. For this experiment, 1-best models were used, since ASR N-best were not available for manual transcripts (i.e., manual transcripts were created by humans and not by ASR). The WER of How2 semi-supervised transcripts was 23.95% and that of standard transcripts was 31.05% (Table 7.1). Table 7.4 summarises the MRR scores of the neural ranking and BERT ranking models applied to manual transcripts, semi-supervised ASR transcripts and standard ASR transcripts. The key points from these results are (i) there is still a gap of search effectiveness between manual and ASR transcripts despite the use of powerful neural ranking and BERT ranking models and (ii) the MRR differences between semi-supervised ASR transcripts and standard ASR transcripts vary depending on the model.

(i) The results show that search effectiveness of all of the models improved when

Table 7.4: MRR results of using manual transcripts, semi-supervised ASR transcripts and standard ASR transcripts for How2 known-item search evaluation.

model	MRR eval data		
	manual	semi-supervised ASR	standard ASR
BM25	40.27	30.85	29.37
DRMM	41.66	32.92	30.44
PACRR	44.80	36.04	33.62
BERT-rerank	57.32	48.69	46.37
BERT-DR	50.61	40.38	34.81
BERT-DR-ANCE	66.73	56.22	52.41

the models were applied to manual transcripts instead of ASR transcripts, indicating that the neural ranking and BERT ranking models could not mitigate the effects of ASR errors by itself. Across all the models, the difference in MRR score between manual transcripts and semi-supervised ASR transcripts was roughly by 10 MRR score. The highest MRR score observed on How2 manual transcripts was 66.73 from the BERT DR model trained using ANCE.

(ii) The MRR differences between semi-supervised ASR transcripts and standard ASR transcripts vary depending on a model. The largest benefit of reduction in WER 7% was brought to the BERT DR system, that improved the MRR score by 5.5 points. The BERT DR system using ANCE training improved the MRR score by 3.8 on semi-supervised ASR transcripts. DRMM, PACRR and BERT-reranking models on semi-supervised ASR transcripts produced around 2.5 MRR score better than standard ASR transcripts counterparts. Overall, improvement of ASR transcripts brought larger gain in MRR to all of the neural ranking and BERT ranking models than to the BM25 model.

7.5.4 MRR results on Blip10000

This section presents MRR results of the known-item search task on Blip10000 data. The Blip10000 corpus is a collection of spoken videos and this task can be seen as evaluation of the neural ranking and BERT ranking models in the out-of-domain scenario where these models were trained on the How2 corpus and evaluated on a

Table 7.5: MRR results of DRMM, PACRR and BERT-DR models using the N-best extension on Blip10000 known-item task. The MRR scores better than the BM25 score are shown in the bold face.

model	standard ASR N-size			semi-supervised ASR N-size		
	1	5	10	1	5	10
BM25	17.15			29.22		
DRMM	4.96	4.71	7.09	11.32	10.07	11.38
PACRR	13.88	**18.16**	**21.97**	22.11	27.01	**29.72**
BERT rerank	**27.87**	–	–	**29.84**	–	–
BERT DR	9.66	11.73	10.82	12.10	18.18	13.37
BERT DR ANCE	13.74	14.45	13.40	19.01	19.21	18.72

different corpus. Two types of transcripts were prepared for the Blip10000 known-item search: standard ASR transcripts and semi-supervised ASR transcripts. The average WER of blip dev and test set of semi-supervised transcripts was 32.75%, while that of standard transcripts was 37.98% (Table 7.1). Similar to the How2 experiments, DRMM, PACRR and BERT re-ranking models were applied to re-ranking the top 1,000 ranked list returned by the BM25 model for each evaluation query. The N-best models were applied to {5, 10}-best hypotheses and late fusion was used for the N-best BERT DR model. The BERT DR model encoded evaluation queries and the whole Blip10000 documents into vectors and ran ANN over each query vector against each document vector in the index to create a ranked list of documents.

Two purposes of this experiment are: (i) to examine the effects of using semi-supervised ASR transcripts instead of standard ASR transcripts on a diverse video corpus and (ii) to investigate abilities of the neural and BERT ranking models to generalise on an out-of-domain corpus. We have shown that the BM25 model can benefit from using semi-supervised ASR transcripts instead of standard transcripts (Table 7.1). This experiment investigates whether the same gain in MRR score can be brought to the neural and BERT ranking models. The second goal of the experiment is to examine whether the neural and BERT ranking models trained on the How2 corpus could be effective on the Blip10000 corpus.

Table 7.5 summarises the known-item search experiments on the Blip10000 corpus. In this experiment, N-best PACRR and BERT re-rank models outperformed the BM25 model. In comparison to How2 experiments, gain in the MRR score brought by neural and BERT models was small except for the BERT re-rank model on the standard ASR transcripts that improved the MRR score by 10 points over the BM25 model. The BERT DR model using ANCE training was effective in the How2 known-item search task, while the Blip results indicate that the DR models was not able to produce good vector representation of queries and documents. Overall, the Blip known-item search results indicate that neural ranking and BERT ranking models need a large amount of training data to generalise on a task of another domain.

Another interesting point of the Blip known-item search result is the use of semi-supervised ASR transcripts. Although most of the neural ranking and BERT ranking models did not produce MRR scores better than BM25, the semi-supervised ASR transcripts seemed to benefit these models. For example, using the semi-supervised ASR transcripts increased the MRR score of DRMM from 4.96 to 11.32, that of PACRR from 13.88 to 22.11, the BERT DR model from 13.74 to 19.01. Using the semi-supervised ASR transcripts did not bring a large MRR improvement to the BERT re-ranking and BERT DR models. Finally, the difficulty of the Blip known-item search task could be increased due to the short length of Blip known-item queries. The average length of How2 evaluation queries was 10.34, while that of Blip1000 queries was 5.82. During the query creation (Section 7.2.2), AMT Workers were asked to provide at least 3 words to find a presented video on video streaming services, and many created queries became short such as “Waikiki beach foodie”, “Nick Dutch tarot reading”

7.6 Discussion and Conclusions

Recently proposed neural ranking models have been shown to be more effective on IR passage and document retrieval tasks the traditional probabilistic ranking models (Guo et al., 2020; Lin et al., 2020). Although BERT-based ranking models have been used in the TREC Podcasts Tracks (Jones et al., 2020, 2021), no manual transcripts were available in the Tracks, and it was not clear whether the neural ranking models could overcome ASR transcription errors. The motivation of investigations in this chapter was to further explore effectiveness and capabilities of neural ranking models in the SCR task. This chapter also introduced the ASR N-best extension of DRMM, PACRR and BERT DR models. The motivation of this extension method was to mitigate ASR transcription errors by using multiple hypotheses from an ASR system rather than 1-best transcripts.

This chapter presented two known-item search experiments using different types of transcripts of the How2 corpus and the Blip10000 corpus. While the How2 experiment was intended to be an in-domain task where ranking models operated on the same domain text during training and evaluation, the Blip10000 experiment was an out-of-domain scenario where model evaluation was conducted on texts of an unseen domain. The goal of these investigations was to examine the difference in search effectiveness between three types of transcripts: manual transcripts, standard ASR transcripts and semi-supervised ASR transcripts and to examine abilities of the neural ranking models to generalise on a corpus of a different domain.

The findings from our experiments are as follows. The How2 experimental results demonstrated that all of the neural ranking and BERT ranking models outperformed the BM25 model and the BERT DR model using ANCE training was the most effective. Further, our N-best extension to DRMM, PACRR and BERT-DR models had a positive impact on the How2 search results. Nevertheless, the How2 experiments also revealed that neural ranking models were much more effective using manual transcripts than standard ASR and semi-supervised ASR transcripts. Another interesting finding in the How2 experiment was search effectiveness of neural models did not differ when train data was manual transcripts or standard ASR transcripts. This indicates that neural models can be trained on pairs of ASR transcripts and video titles, removing the need to create costly manual transcripts of spoken documents. The results from Blip10000 showed a weakness of neural models, in that these models require a large amount of train data to be effective on out-of-domain data. The BERT re-ranking model using the standard ASR transcripts was the only model that brought large MRR improvement over BM25.

Chapter 8

Conclusions and Future Work

This thesis conducted research investigations to improve ASR transcripts for SCR and SCR effectiveness for a collection that is likely to satisfy a user's information needs. The particular challenges lying in SCR tasks relate to presence of ASR transcription errors in transcripts of spoken documents. Such errors can prevent ranking systems from correctly matching a user query with spoken documents in which users are interested. Especially, transcription errors of "keywords" including proper nouns and entities can have a negative impact on search effectiveness. Although recent years have seen significant improvements in ASR systems, recognition accuracy of ASR systems can still drop when ASR is used to recognise speech with highly varied speaker characteristics, speaking styles and acoustic conditions. These issues motivated us to conduct three research investigations: (i) integration of multimodal information into ASR to correctly transcribe nouns and entities, (ii) semi-supervised training for ASR to improve transcription accuracy of a collection of highly varied spoken documents and (iii) augmenting SCR ranking models using N-best ASR hypotheses to mitigate ASR transcription errors.

The final chapter of this thesis summarises the findings of the research investigations conducted in this thesis. The research questions proposed in Section 2.4 and Section 3.5 are re-visited and these research questions are answered with respect to the experimental results of this thesis. The final section of this chapter provides future directions of SCR research on highly varied spoken document collections.

8.1 Summary of Findings from Research Investigations

This thesis conducted three research investigations into ASR and SCR systems. Chapter 5 focused on the use of multimodal information into ASR systems. The motivation of this method was to correctly transcribe “keywords” that are important for search operations. Chapter 6 investigated the use of semi-supervised training for ASR systems for transcription of highly varied spoken documents. Since ASR systems are often developed and evaluated on a corpus of the same domain, recognition accuracy drops when these systems are used for transcription of out-of-domain data. We used semi-supervised training with a potential to resolve an issue of transcribing domain mismatch data. Chapter 7 examined the use of neural ranking and BERT ranking models and the extension of these models using ASR N-best hypotheses. The reminder of this section summarises the research findings from each research investigation.

8.1.1 Findings from multimodal augmentation of ASR systems

Human speech processing is fundamentally multimodal and situational contexts are constantly updated to correctly understand speech. This motivated our investigation into the use of multimodal information for ASR by integrating background situational contexts into ASR systems, as standard ASR systems generally work on speech without exploiting the situational contexts. The three multimodal features investigated were: visual context features, video title features and speaker face features. While visual context and video title features were incorporated into RNN language models to re-score ASR N-best hypotheses, speaker face features were integrated into a neural acoustic model to exploit visual information about speaker characteristics.

Experiments using multimodal features were carried out on the How2 corpus consisting of instruction videos and on the Udacity corpus containing lecture videos. Compared to the baseline systems without using multimodal features, all of the multimdoal features used contributed to a small gain in WERs on both corpora. Analysis of visual context and video title features might indicate that multimodally augmented RNN language models predict nouns and entities with higher confidence. Based also on analysis, the acoustic model might be able to capture the relationship between visual speaker attributes and speech from the speaker face feature. Further research could be conducted to identify multimodal features being able to provide useful information to acoustic and language models.

8.1.2 Findings from semi-supervised training and acoustic model adaptation using content genre for ASR

Chapter 6 investigated the use of semi-supervised training and content genre information for ASR systems as a potential solution to recognise speech in out-of-domain data. This is important for SCR tasks because ASR systems are often used to transcribe a collection of spoken documents with highly varied speaker characteristics and content domains. While the use of semi-supervised training was motivated to exploit untranscribed in-domain data for ASR training, the use of content genre was proposed to mitigate the effects of particular types of acoustic noise more likely to occur in a certain genre and to enable ASR to handle such noise. The chapter examined the whole pipeline process to prepare raw unsegmented speech data for semi-supervised training and optimised semi-supervised methods for acoustic models, N-gram models and RNN language models.

In the semi-supervised experiments, the seed ASR system was developed using the How2 corpus and the augmented ASR systems were evaluated on the Blip10000 corpus containing highly varied spoken documents. Overall, the whole optimised process of semi-supervised training and content genre adaptation brought around 5% WER improvement. The largest impact on WER improvement was obtained by semi-supervised training leading to 1.5-2% WER reduction and the RNN lattice re-scoring also leading to 1.5-2% WER reduction. Content genre adaptation of a neural acoustic model and semi-supervised training of N-gram and RNN language models using 1-best transcripts from the seed system led to around 0.5% WER gain each. The smallest effect on WER was re-scoring decoder lattices using the seed RNN language model on which a semi-supervised acoustic model was trained. This brought around 0.3% WER improvement. Semi-supervised ASR transcripts created in this investigation were used for the SCR task in Chapter 7 in comparison with standard ASR transcripts.

8.1.3 Findings from augmentation of SCR ranking methods

Recent years have seen active research on ranking models using neural networks and transformer architecture that have been demonstrated to be more effective than the traditional probabilistic ranking models. Although ranking models using transformer architecture were employed in the past TREC Podcasts Tracks (Jones et al., 2020, 2021), due to the absence of manual transcripts in the tasks, it is not clear whether these models overcome transcription errors present in ASR transcripts. It is further motivated to augment neural ranking and transformer ranking models with the use of ASR N-best hypotheses to mitigate the effects of transcription errors. Chapter 7 compared manual transcripts, semi-supervised ASR transcripts and standard ASR transcripts using these ranking models with respect to search effectiveness and examined the benefits of using N-best hypotheses for these neural ranking models.

The SCR experiments were conducted on the How2 corpus as an in-domain scenario where search models were trained and evaluated on the same domain data and on the Blip10000 corpus as an out-of-domain scenario where evaluation was conducted on domain mis-matched data. Experimental results indicate that transformer ranking models, especially the BERT DR model using ANCE training, were highly effective on the How2 known-item search. Nevertheless, all of the neural ranking models produced around 10 points higher MRR on manual transcripts than on semi-supervised ASR transcripts, indicating that these models are not able to overcome transcription errors without modifications. The effects of ASR errors on DRMM, PACRR and BERT DR models were somewhat mitigated by the ASR N-best extension, increasing 2-4 MRR score on the How2 experiment. Finally, only BERT re-ranking and 10-best PACRR outperformed the BM25 model in the Blip10000 experiment, indicating that a larger scale dataset is required for generalisation of the neural ranking models.

8.2 Re-visiting Research Questions

Four research questions (RQs) relating to ASR and SCR investigations were proposed in Section 2.4 and Section 3.5. This section reviews these research questions and provides answers with regard to experimental findings found in this thesis.

RQ1: Can incorporation of visual features be used for improvement of ASR transcription accuracy?

Our experimental results in Chapter 5 demonstrated marginal improvement of WERs using visual context features, video title features and speaker face features. Practical WER improvement might be achieved if the scale of the training dataset could be increased for ASR to learn more generalised relationships between speech and situational contexts.

RQ2: Can semi-supervised training and acoustic model adaptation improve ASR accuracy for content from diverse uncontrolled topical domains?

Our experiments in Chapter 4 showed that transcription accuracy obtained by the whole process of semi-supervised training and content genre adaptation was around 5% WER. The semi-supervised ASR transcripts were demonstrated to be more useful for the search task in Chapter 7 than standard ASR transcripts. Semi-supervised training and adaptation techniques showed a potential to improve ASR accuracy for diverse spoken content and these techniques can be refined to obtain further improvement of transcription accuracy.

RQ3: How effective are neural ranking and transformer ranking models for SCR?

Experimental results in Chapter 7 provides a good indication that neural ranking and transformer ranking models are effective for SCR tasks. In particular, the BERT DR model using ANCE training produced around a 26 point higher MRR score than the BM25 model. However, most of the neural ranking and BERT ranking models struggled for the known-item search task of Blip10000 that was an out-of-domain scenario. Further, a gap in MRR scores between manual transcripts and semi-supervised ASR transcripts was still around 10 points despite the use of neural ranking and BERT ranking models.

RQ4: Can current neural models be extended for improved SCR effectiveness, e.g. using N-best ASR hypotheses to increase search effectiveness

The neural ranking models successfully improved the MRR scores by around 2-4 points using N-best ASR hypotheses in experimental results shown in Chapter 7. Search effectiveness could be further improved by using decoder lattices rather than ASR N-best hypotheses, since the lattices contain even richer information about decoded spoken documents than N-best hypotheses that are simple text representations.

8.3 Future SCR Research on Highly Varied Spoken Document Collections

Recent years have seen an explosion of information with the rapid increase in the amount of multimedia content, including user-generated videos and podcasts, available on the Internet. This has occurred due to the availability of popular content sharing services such as YouTube and Spotify Podcasts. Spoken multimedia content is generally searched by matching a user query with user provided content titles, descriptions or click logs. For more sophisticated SCR features, it is desirable to have accurate speech transcripts of spoken documents and analyse what is spoken and who is speaking.

Despite development of sophisticated ranking models using BERT architecture, experimental findings in this thesis showed that ASR transcription errors still have a negative impact on search effectiveness of these ranking models. Two solutions to resolve this are: improve ASR transcription accuracy on diverse spoken content or enable ranking models to be robust to transcription errors. To achieve this, the following future research directions of SCR are proposed.

8.3.1 Creation of diverse data with manual transcripts available

Evaluation of ranking models is an indispensable process to measure effectiveness of these models. Although very large scale SCR tasks were conducted in the TREC Podcasts Tracks (Jones et al., 2020, 2021), it was not feasible to create manual transcripts of over 50,000 hours of data. In the past SCR challenges, transcripts of closed caption quality were available in the TREC Spoken Document Retrieval (SDR) Tracks (Garofolo et al., 2000) and the MediaEval Search and Hyperlinking tasks 2013-2014 (Eskevich et al., 2013a, 2014). The first proposal to further SCR research is to create a very large scale dataset with which closed captions created for accessibility are available. This would allow us to examine the real gap in search effectiveness of sophisticated ranking models such as BERT re-ranking over manual transcripts and ASR transcripts.

8.3.2 Context-aware ASR systems

This thesis conducted research on the integration of multimodal information into ASR. Although improvement of recognition accuracy was marginal, multimodally augmented systems were demonstrated to be aware of the situational contexts provided by the multimodal features. Two proposals to further improve context awareness of ASR systems for transcription of "keywords" are: the use of content description and the use of 1st-pass decode to refine transcripts.

In this thesis, video titles were used for augmentation of RNN language models and content genre information was used for neural acoustic models. In addition to these types of metadata, content descriptions are often made available in multimedia sharing services. Content descriptions might be even better information sources of situational contexts of spoken documents and integration of these descriptions into language models might help to create more accurate transcripts.

Another proposal to ASR augmentation is the use of 1st-pass decode output. Unlike ASR systems used for dictation or personal assistants, SCR systems do not require ASR transcripts to be available immediately after receiving speech data. The proposal here is to use 1st-pass decode output to collect broad information about decoded spoken content and exploit this information for refinement of transcripts in the 2nd-pass decode.

8.3.3 Noise robust BERT ranking models

In this thesis, BERT models showed superior search effectiveness over other neural ranking models and the BM25 model in the How2 experiment. Aside from the obvious challenges of generalising these ranking models, this section proposes two augmentation of these ranking models to be robust to ASR transcription errors: resolving length limitation of BERT ranking models and using ASR decoder lattices as input of ranking models.

The first proposal is to resolve the length limitation of BERT (Boytsov and Kolter, 2021; Dai and Callan, 2019). Chapter 3 described BERT architecture and discussed that the limit of an input sequence for BERT is often 512 tokens. In other words, documents longer than this limit are truncated. This is a limitation to SCR tasks, since ASR transcripts can be longer than the pre-defined length limit. This motivates us to develop an approach to resolving the length limit of BERT or to create an effective ranking model that is more flexible to document lengths similar to DRMM.

The second proposal is to use a richer representation of spoken documents for BERT ranking models. This thesis demonstrated that the use of N-best ASR hypotheses could mitigate the ASR transcription errors contained in the transcripts. ASR N-best transcripts, however, do not keep all of the possible hypotheses retained during decoding and scores from an acoustic model and an N-gram language model are removed. The lattice format, on the other hand, retains all of the above information discarded by the N-best transcripts. Thus, using the decoder lattices as input of BERT ranking models might further mitigate ASR transcription errors.

Bibliography

Abdel-Hamid, O. and Jiang, H. (2013). Fast speaker adaptation of hybrid nn/hmm model for speech recognition based on discriminative learning of speaker code. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7942–7946.

Akiba, T., Nishizaki, H., Aikawa, K., Hu, X., Itoh, Y., Kawahara, T., Nakagawa, S., and Nanjo, H. (2013). Overview of the ntcir-10 spokendoc-2 task. In *Proceedings of the 10th NTCIR Conference on Evaluation of Information Access*, pages 573–587.

Akiba, T., Nishizaki, H., Aikawa, K., Kawahara, T., and Tomoko, M. (2011). Overview of the ir for spoken documents task in ntcir-9 workshop. In *Proceedings of the 9th NTCIR Conference on Evaluation of Information Access*, pages 223–235.

Arora, S., Liang, Y., and Ma, T. (2017). A simple but tough to beat baseline for sentence embeddings. *Proceedings of the International Conference on Learning Representations (ICLR).*

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR).*

Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., et al. (2016). MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268.*

Bell, P., Gales, M. J. F., Hain, T., Kilgour, J., Lanchantin, P., Liu, X., McParland, A., Renals, S., Saz, O., Wester, M., and Woodland, P. C. (2015). The MGB challenge: Evaluating multi-genre broadcast media recognition. In *Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 687–693.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Boytsov, L. and Kolter, Z. (2021). Exploring classic and neural lexical translation models for information retrieval: Interpretability, effectiveness, and efficiency benefits. In *Advances in Information Retrieval: the European Conference on IR Research (ECIR)*, page 63–78.

Caglayan, O., Sanabria, R., Palaskar, S., Barraul, L., and Metze, F. (2019). Multimodal grounding for sequence-to-sequence speech recognition. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8648–8652.

Carmantini, A., Bell, P., and Renals, S. (2019). Untranscribed Web Audio for Low Resource Speech Recognition. In *Proceedings of Interspeech 2019*, pages 226–230.

Çelebi, A. and Saraçlar, M. (2013). Semi-supervised discriminative language modeling with out-of-domain text data. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 727–732.

Chelba, C., Silva, J., and Acero, A. (2007). Soft indexing of speech content for search in spoken documents. *Computer Speech & Language*, 21(3):458–478.

Chen, S. F. and Goodman, J. (1995). An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, pages 310–318.

Chen, Y.-C., Huang, S.-F., Shen, C.-H., Lee, H.-y., and Lee, L.-s. (2018). Phonetic-and-semantic embedding of spoken words with applications in spoken content retrieval. In *Proceedings of Spoken Language Technology Workshop (SLT)*, pages 941–948.

Craswell, N., Mitra, B., Yilmaz, E., Campos, D., and Lin, J. (2021). Overview of the trec 2021 deep learning track. In *Proceedings of Text REtrieval Conference (TREC)*.

Dai, Z. and Callan, J. (2019). Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 985–988.

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366.

Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Ouellet, P. (2011). Front-end factor analysis for speaker verification. *Transactions on Audio, Speech, and Language Processing*, 19(4):788–798.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 4171–4186.

Dharanipragada, S., Franz, M., and Roukos, S. (1998). Audio-indexing for broadcast news. In *Proceedings of Text REtrieval Conference (TREC-7)*, pages 63–67.

Eskevich, M., Aly, R., Ordelman, R., Chen, S., and Jones, G. J. (2013a). The search and hyperlinking task at mediaeval 2013. In *Proceedings of MediaEval*.

Eskevich, M., Aly, R., Racca, D., Ordelman, R., Chen, S., and Jones, G. J. (2014). The search and hyperlinking task at mediaeval 2014. In *Proceedings of MediaEval*.

Eskevich, M. and Jones, G. J. (2013). Time-based segmentation and use of jum-in points in dcu search runs at the search and hyperlinking task at mediaeval 2013. In *Proceedings of MediaEval*.

Eskevich, M., Jones, G. J., Chen, S., Aly, R., Ordelman, R., and Larson, M. (2012). Search and hyperlinking task at mediaeval 2012. In *Proceedings of MediaEval*.

Eskevich, M., Jones, G. J. F., Aly, R., Ordelman, R. J., Chen, S., Nadeem, D., Guinaudeau, C., Gravier, G., Sébillot, P., De Nies, T., Debevere, P., Van De Walle, R., Galuscakova, P., Pecina, P., and Larson, M. (2013b). Multimedia Information Seeking through Search and Hyperlinking. In *Proceedings of the International Conference on Multimedia Retrieval (ICMR)*.

Fainberg, J., Renals, S., and Bell, P. (2017). Factorised representations for neural network adaptation to diverse acoustic environments. In *Proceedings of Interspeech 2017*, pages 749–753.

Feng, X., Richardson, B., Amman, S., and Glass, J. (2017). An environmental feature representation for robust speech recognition and for environment identification. In *Proceedings of Interspeech*, pages 3078–3082.

Fleischman, M. and Roy, D. (2008). Grounded language modeling for automatic speech recognition of sports video. In *Proceedings of the Association for Computational Linguistics (ACL-HLT)*, pages 121–129.

Gales, M. and Young, S. (2008). The application of hidden markov models in speech recognition. *Foundations and Trends in Signal Processing*, 1(3):195–304.

Galuscakova, P., Nair, S., and Oard, D. W. (2020). Combine and re-rank: The university of maryland at the TREC 2020 podcasts track. In *Proceedings of Text REtrieval Conference (TREC)*, volume 1266.

Ganesan, K., Bamdev, P., B, J., Venugopal, A., and Tushar, A. (2021). N-best ASR transformer: Enhancing SLU performance using multiple ASR hypotheses. In *Proceedings of Association for Computational Linguistics (ACL)*, pages 93–98.

Garofolo, J. S., Auzanne, C. G. P., and Voorhees, E. M. (2000). The TREC spoken document retrieval track: A success story. In *Content-Based Multimedia Information Access - Volume 1*, page 1–20.

Garofolo, J. S., Voorhees, E. M., Auzanne, C. G. P., Stanford, V. M., and Lund, B. A. (1998). 1998 TREC-7 spoken document retrieval track overview and results. In *Proceedings of the Text Retrieval Conference (TREC-7).*

Garofolo, J. S., Voorhees, E. M., Stanford, V. M., and Sparck Jones, K. (1997). TREC-6 1997 spoken document retrieval track overview and results. In *Proceedings of the Text Retrieval Conference (TREC-6)*, pages 83–91.

Guo, J., Fan, Y., Ai, Q., and Croft, W. B. (2016). A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the International Conference on Information and Knowledge Management*, page 55–64. Association for Computing Machinery.

Guo, J., Fan, Y., Pang, L., Yang, L., Ai, Q., Zamani, H., Wu, C., Croft, W. B., and Cheng, X. (2020). A deep look into neural ranking models for information retrieval. *Information Processing & Management*, 57(6):102067.

Gupta, A., Miao, Y., Neves, L., and Metze, F. (2017). Visual Features for Context-aware Speech Recognition. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5020–5024.

Hadian, H., Sameti, H., Povey, D., and Khudanpur, S. (2018). End-to-end speech recognition using lattice-free mmi. In *Proceedings of Interspeech*, pages 12–16.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

Hinton, G. and S. Osindero, S. Y.-W. T. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–80.

Hofstatter, S., Sertkan, M., and Hanbury, A. (2021). TU wien at TREC DL and podcast 2021: Simple compression for dense retrieval. In *Proceedings of Text REtrieval Conference (TREC)*.

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. (2013a). Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the International Conference on Information and Knowledge Management*, page 2333–2338.

Huang, Y., Yu, D., Gong, Y., and Liu, C. (2013b). Semi-supervised GMM and DNN acoustic model training with multi-system combination and confidence re-calibration. In *Proceedings of Interspeech*, pages 2360–2364.

Hui, K., Yates, A., Berberich, K., and de Melo, G. (2017). PACRR: A position-aware neural IR model for relevance matching. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1049–1058. Association for Computational Linguistics.

James, D. A. and Young, S. (1994). A fast lattice-based approach to vocabulary independent wordspotting. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume i, pages 377–380.

Jelinek, F. (1976). Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64(4):532–556.

Johnson, J., Douze, M., and Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Johnson, S., Jourlin, P., Moore, G., Sparck Jones, K., and Woodland, P. (1998). Spoken document retrieval for TREC-7 at Cambridge University. In *Proceedings of Text REtrieval Conference (TREC-7)*, pages 138–147.

Johnson, S., Jourlin, P., Sparck Jones, K., and Woodland, P. (1999a). Spoken document retrieval for TREC-8 at Cambridge University. In *Proceedings of Text REtrieval Conference (TREC-8)*.

Johnson, S. J., Jourlin, P., Moore, G. L., Spärck Jones, K., and Woodland, P. (1999b). The cambridge university spoken document retrieval system. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 49–52.

Jones, R., Cartere, B., Clifton, A., Eskevich, M., Jones, G. J. F., Karlgren, J., Pappu, A., Reddy, S., and Yu, Y. (2020). TREC 2020 podcasts track overview. In *Proceedings of Text REtrieval Conference (TREC)*.

Jones, R., Cartere, B., Clifton, A., Eskevich, M., Jones, G. J. F., Karlgren, J., Pappu, A., Reddy, S., and Yu, Y. (2021). TREC 2021 podcasts track overview. In *Proceedings of Text REtrieval Conference (TREC)*.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Kim, S., Raj, B., and Lane, I. (2016). Environmental noise embeddings for robust speech recognition. *arXiv preprint arXiv:1601.02553*.

King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758.

Lamel, L. and Gauvain, J.-L. (2008). Speech processing for audio indexing. In *Advances in Natural Language Processing*, pages 4–15.

Larson, M. and Jones, G. J. F. (2012). Spoken Content Retrieval: A survey of techniques and technologies. *Foundations and Trends in Information Retrieval*, 4(4-5):235–422.

Lee, K. (1990). Context-independent phonetic hidden markov models for speaker-independent continuous speech recognition. *Transactions on Acoustics, Speech, and Signal Processing*, 38(4):599–609.

Lin, J., Nogueira, R., and Yates, A. (2020). Pretrained transformers for text ranking: BERT and beyond. *arXiv preprint arXiv:2010.06467*.

Liu, X., Wang, Y., Chen, X., Gales, M. J. F., and Woodland, P. C. (2014). Efficient lattice rescoring using recurrent neural network language models. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4908–4912.

Lüscher, C., Beck, E., Irie, K., Kitza, M., Michel, W., Zeyer, A., Schlüter, R., and Ney, H. (2019). RWTH ASR Systems for LibriSpeech: Hybrid vs Attention. In *Proceedings of Interspeech*, pages 231–235.

Majumdar, S. and Ginsburg, B. (2020). MatchboxNet: 1D Time-Channel Separable Convolutional Neural Network Architecture for Speech Commands Recognition. In *Proceedings of Interspeech 2020*, pages 3356–3360.

Mamou, J., Carmel, D., and Hoory, R. (2006). Spoken document retrieval from call-center conversations. In *Proceedings of Special Interest Group on Information Retrieval (SIGIR)*, page 51–58.

Manohar, V., Hadian, H., Povey, D., and Khudanpur, S. (2018). Semi-supervised training of acoustic models using lattice-free mmi. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4844–4848.

McDonald, R., Brokos, G., and Androutsopoulos, I. (2018). Deep relevance ranking using enhanced document-query interactions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1849–1860.

McGurk, H. and MacDonald, J. (1976). Hearing lips and seeing voices. *Nature*, 264(5588):746–748.

Miao, Y., Jiang, L., Zhang, H., and Metze, F. (2014). Improvements to speaker adaptive training of deep neural networks. In *Proceedings of Spoken Language Technology Workshop (SLT)*, pages 165–170.

Mikolov, T., Karafiat, M., Burget, L., Cernocky, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Proceedings of Interspeech*, pages 1045–1048.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119.

Mikolov, T. and Zweig, G. (2012). Context dependent recurrent neural network language model. In *Proceedings of Spoken Language Technology Workshop (SLT)*, pages 234–239.

Mohamed, A.-r., Dahl, G. E., and Hinton, G. (2012). Acoustic modeling using deep belief networks. *Transactions on Audio, Speech, and Language Processing*, 20(1):14–22.

Mohri, M., Pereira, F., and Riley, M. (2002). Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88.

Moriya, Y. and Jones, G. J. F. (2021a). An ASR N-best transcript neural ranking model for spoken content retrieval. In *In Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU).*

Moriya, Y. and Jones, G. J. F. (2021b). Augmenting asr for user-generated videos with semi-supervised training and content genreadaptation. In *Proceedings of Statistical Language and Speech Processing (SLSP) (under review).*

Narayanan, A., Misra, A., Sim, K. C., Pundak, G., Tripathi, A., Elfeky, M., Haghani, P., Strohman, T., and Bacchiani, M. (2018). Toward domain-invariant speech recognition via large scale training. In *Proceedings of Spoken Language Technology Workshop (SLT)*, pages 441–447.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 689–696.

Nogueira, R. and Cho, K. (2019). Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085.*

Oard, D. W., Wang, J., Jones, G. J. F., White, R. W., Pecina, P., Soergel, D., Huang, X., and Shafran, I. (2007). Overview of the clef-2006 cross-language speech retrieval track. In Peters, C., Clough, P., Gey, F. C., Karlgren, J., Magnini, B., Oard, D. W., de Rijke, M., and Stempfhuber, M., editors, *Evaluation of Multilingual and Multi-modal Information Retrieval*, pages 744–758.

Oba, T., Ogawa, A., Hori, T., Masataki, H., and Nakamura, A. (2013). Unsupervised discriminative language modeling using error rate estimator. In *Proceedings of Interspeech*, pages 1223–1227.

Palaskar, S., Sanabria, R., and Metze, F. (2018). End-to-End Multimodal Speech Recognition. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5774–5778.

Pallett, D. S., Fiscus, J., Garofolo, J., Martin, A., and Przybocki, M. (1999). 1998 broadcast news benchmark test results: English and non-english word error rate performance measures. In *DARPA Broadcast News Transcription and Understanding Workshop.*

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. In *British Machine Vision Conference*.

Paszke, A., Chanan, G., Lin, Z., Gross, S., Yang, E., Antiga, L., and Devito, Z. (2017). Automatic differentiation in PyTorch. *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 1–4.

Pecina, P., Hoffmannová, P., Jones, G. J. F., Zhang, Y., and Oard, D. W. (2008). Overview of the clef-2007 cross-language speech retrieval track. In *Advances in Multilingual and Multimodal Information Retrieval*, pages 674–686.

Peddinti, V., Povey, D., and Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *Proceedings of Interspeech*, pages 2–6.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Potamianos, G., Neti, C., Gravier, G., Garg, A., and Senior, A. W. (2003). Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The Kaldi speech recognition toolkit. In *Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 1–4.

Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., and Khudanpur, S. (2016). Purely sequence-trained neural networks for asr based on lattice-free mmi. In *Proceedings of Interspeech*, pages 2751–2755.

Racca, D. N., Eskevich, M., and Jones, G. J. (2014). Dcu search runs at mediaeval 2014 search and hyperlinking. In *Proceedings of MediaEval*.

Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. (1995). Okapi at TREC-3. In *Proceedings of Text REtrieval Conference (TREC)*, pages 109–123. National Institute of Standards and Technology.

Robertson, S. E. and Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the International SIGIR Conference*, pages 232–241.

Rousseau, A., Bougares, F., Deléglise, P., Schwenk, H., and Estève, Y. (2011). LIUM's systems for the IWSLT 2011 speech translation tasks. In *Proceedings of the International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 79–85.

Roy, B. C., Vosoughi, S., and Roy, D. (2014). Grounding language models in spatiotemporal context. In *Proceedings of Interspeech*, pages 2625–2629.

Rumelhart, D., Hinton, G. E., and Williams, J. R. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Sainath, T. N., He, Y., Li, B., Narayanan, A., Pang, R., Bruguier, A., Chang, S.-y., Li, W., Alvarez, R., Chen, Z., Chiu, C.-C., Garcia, D., Gruenstein, A., Hu, K., Kannan, A., Liang, Q., McGraw, I., Peyser, C., Prabhavalkar, R., Pundak, G., Rybach, D., Shangguan, Y., Sheth, Y., Strohman, T., Visontai, M., Wu, Y., Zhang, Y., and Zhao, D. (2020). A streaming on-device end-to-end model surpassing server-side conventional model quality and latency. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6059–6063.

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

Sanabria, R., Caglayan, O., Palaskar, S., Elliott, D., Barrault, L., Specia, L., and Metze, F. (2018). How2: a large-scale dataset for multimodal language understanding. In *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*. NeurIPS.

Sanderson, M. and Croft, W. B. (2012). The history of information retrieval research. *Proceedings of the IEEE*, 100(Special Centennial Issue):1444–1451.

Sanderson, M. and Shou, X. M. (2007). Search of spoken documents retrieves well recognized transcripts. In Amati, G., Carpineto, C., and Romano, G., editors, *Advances in Information Retrieval*, pages 505–516. Springer Berlin Heidelberg.

Saon, G., Soltau, H., Nahamoo, D., and Picheny, M. (2013). Speaker adaptation of neural network acoustic models using i-vectors. In *Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 55–59.

Schmiedeke, S., Xu, P., Ferrané, I., Eskevich, M., Kofler, C., Larson, A. M., Estève, Y., Lamel, L., Jones, G. J. F., and Sikora, T. (2013). Blip10000: a social video dataset containing spug content for tagging and retrieval. In *Proceedings of ACM MMSys 2013*.

Sheikh, I., Vincent, E., and Illina, I. (2020). On Semi-Supervised LF-MMI Training of Acoustic Models with Limited Data. In *Proceedings of Interspeech*, pages 986–990.

Shou, X. M., Sanderson, M., and Tuffs, N. (2003). The relationship of word error rate to document ranking. In *Proceedings of the AAAI Spring Symposium on Intelligent Multimedia Knowledge Management*, pages 28–33.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *Proceedings of International Conference on Learning Representations (ICLR).*

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. (2018). X-vectors: Robust DNN embeddings for speaker recognition. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333.

Stolcke, A. (2002). SRILM-an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP).*

Su, H. and Xu, H. (2015). Multi-softmax deep neural network for semi-supervised training. In *Proceedings of Interspeech 2015.*

Tanenhaus, M., Spivey-Knowlton, M., Eberhard, K., and Sedivy, J. (1995). Integration of visual and linguistic information in spoken language comprehension. *Science*, 268(5217):1632–1634.

Thomas, S., Suzuki, M., Huang, Y., Kurata, G., Tuske, Z., Saon, G., Kingsbury, B., Picheny, M., Dibert, T., Kaiser-Schatzlein, A., and Samko, B. (2019). English broadcast news speech recognition by humans and machines. In *Proceedings of the*

International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6455–6459.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Veselỳ, K., Hannemann, M., and Burget, L. (2013). Semi-supervised training of deep neural networks. In *Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 267–272.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

White, R. W., Oard, D. W., Jones, G. J. F., Soergel, D., and Huang, X. (2006). Overview of the clef-2005 cross-language speech retrieval track. In Peters, C., Gey, F. C., Gonzalo, J., Müller, H., Jones, G. J. F., Kluck, M., Magnini, B., and de Rijke, M., editors, *Accessing Multilingual Information Repositories*, pages 744–759.

Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. (2016). Towards Universal Paraphrastic Sentence Embeddings. *Proceedings of International Conference on Learning Representations (ICLR).*

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pages 38–45. Association for Computational Linguistics.

Xiong, L., Xiong, C., Li, Y., Tang, K.-F., Liu, J., Bennett, P. N., Ahmed, J., and Overwijk, A. (2021). Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations (ICLR).*

Xu, H., Chen, T., Gao, D., Wang, Y., Li, K., Goel, N., Carmiel, Y., Povey, D., and Khudanpur, S. (2018). A pruned Rnnlm lattice-rescoring algorithm for automatic speech recognition. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5929–5933.

Young, S. J., Odell, J. J., and Woodland, P. C. (1994). Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 307–312.

Yu, K., Gales, M., Wang, L., and Woodland, P. C. (2010). Unsupervised training and directed manual transcription for lvcsr. *Speech Communication*, 52(7):652–663.

Zhu, L., Liu, W., Liu, L., and Lin, E. (2021). Improving ASR error correction using n-best hypotheses. In *Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 83–89.