UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

Automatic Speech Recognition Quality Estimation

Shahab Jalalvand

Advisor:

Daniele Falavigna

Fondazione Bruno Kessler

Co-Advisors:

Dr. Marco Turchi

Matteo Negri

Fondazione Bruno Kessler

April 2016

Abstract

Evaluation of automatic speech recognition (ASR) systems is difficult and costly, since it requires manual transcriptions. This evaluation is usually done by computing word error rate (WER) that is the most popular metric in ASR community. Such computation is doable only if the manual references are available, whereas in the real-life applications, it is a too rigid condition. A reference-free metric to evaluate the ASR performance is *confidence measure* which is provided by the ASR decoder. However, the confidence measure is not always available, especially in commercial ASR usages. Even if available, this measure is usually biased towards the decoder. From this perspective, the confidence measure is not suitable for comparison purposes, for example between two ASR systems.

These issues motivate the necessity of an automatic quality estimation system for ASR outputs. This thesis explores ASR quality estimation (ASR QE) from different perspectives including: feature engineering, learning algorithms and applications.

From feature engineering perspective, a wide range of features extractable from input signal and output transcription are studied. These features represent the quality of the recognition from different aspects and they are divided into four groups: signal, textual, hybrid and word-based features.

From learning point of view, we address two main approaches: *i)* QE via regression, suitable for single hypothesis scenario; *ii)* QE via machine-learned ranking (MLR), suitable for multiple hypotheses scenario. In the former, a regression model is used to predict the WER score of each single hypothesis that is created through a single automatic transcription channel. In the latter, a ranking model is used to predict the order of multiple hypotheses with respect to their quality. Multiple hypotheses are mainly generated by several ASR systems or several recording microphones.

From application point of view, we introduce two applications in which ASR QE makes salient improvement in terms of WER: *i)* QE-informed data selection for acoustic model adaptation; *ii)* QE-informed system combination. In the former, we exploit single hypothesis ASR QE methods in order to select the best adaptation data for upgrading the acoustic model. In the latter, we exploit multiple hypotheses ASR QE methods to rank and combine the automatic transcriptions in a supervised manner.

The experiments are mostly conducted on CHiME-3 English dataset. CHiME-3 consists of Wall Street Journal utterances, recorded by multiple far distant microphones in noisy environments. The results show that QE-informed acoustic model adaptation leads to 1.8% absolute WER reduction and QE-informed system combination leads to 1.7% absolute WER reduction in CHiME-3 task.

The outcomes of this thesis are packed in the frame of an open source toolkit named TranscRater[1] (transcription rating toolkit) which has been developed based on the aforementioned studies. TranscRater can be used to extract informative features, train the QE models and predict the quality of the reference-less recognitions in a variety of ASR tasks.

Keywords

automatic speech recognition, quality estimation, acoustic model adaptation, system combination

[1] https://github.com/hlt-mt/TranscRater

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Automatic speech recognition (ASR) has been widely studied during the last decades (Rabiner and Juang, 1993; Jelinek, 1997; Huang et al., 2001; Hinton et al., 2012). A lot of work has been done to overcome the barriers of large vocabulary, language/speaker/accent variation, noise, etc (Zhan and Waibel, 1997; Wölfel and McDonough, 2009; Kermorvant, 1999; Jalalvand et al., 2012). Consequent achievements have made ASR systems part of many real-life applications. Dictation system, voice question answering and speech translation are some of these applications. The increased usage of ASR systems demands for suitable procedures for evaluating their performance.

The most common method to evaluate ASR outputs is measuring word error rate (WER), with respect to the manual references (Jelinek, 1997). This measure is computed by counting the number of errors in the recognized string divided by the number of words in the reference.

$$WER = \frac{\#substitutions + \#insertions + \#deletions}{\#referenceWords}$$

Besides all the proficiencies that have made this measure the most popular one in the ASR society, WER has several limitations:

- the manual reference is not always available;

- providing manual reference is costly and
- WER is not always informative about the understandability of an automatic transcription.

For example, if ASR system A recognizes a speech signal as *"he comes from home"* and system B recognizes as *"he comes from horse"*, while the reference is *"he comes from house"*, then the WER of both systems A and B will be equal to 0.25%. Whereas the quality of the two systems are truly different. In fact, Wang et al. (2003a) shows that sometimes even though the WER increases, the speech understanding measure improves. On speech translation framework, Ruiz and Federico (2015) encounter some deficiencies of WER computation implementations. They introduce POWER (phonetically-oriented word error rate) as a variation of WER which incorporates the phonetic alignment as well.

When the manual reference is not available, one can measure the quality of recognition by considering the *confidence measure* which is provided by the decoder (Evermann and Woodland, 2000; Wessel et al., 2001; Seigel, 2013). However, there are two main restrictions with confidence measure. First, not all the systems provide confidence measures (specially the commercial ones). Second, confidence measure is highly biased towards the decoder. That is, sometimes the decoder provides a very high confidence value, even though the hypothesis is wrong. In this way, unsupervised comparison between two ASR systems using their confidence scores will be unreliable. Many researchers tried to increase the reliability of confidence measure. Seigel (2013) uses conditional random field models with word/sub-word features and deletion detection techniques to improve the reliability of the confidence measure. Nevertheless, the resulting confidence score is still highly dependent on the inner behaviour of ASR decoder and from this point of view, unsupervised comparison between heterogeneous

systems, like traditional GMM-HMM[1] versus hybrid DNN-HMM[2] is not trivial.

This thesis explores ASR quality estimation (ASR QE) from different perspectives including:

- feature engineering,
- learning algorithms and
- applications.

Feature engineering. A wide range of features extractable from input signal and output transcription are studied. These features represent the quality of the recognition from different aspects and they are divided into four groups: signal, textual, hybrid and word-based features.

Learning algorithms. Two major strategies are conducted in this thesis: *i)* QE via regression, suitable for single hypothesis scenario; *ii)* QE via machine-learned ranking (MLR), suitable for multiple hypotheses scenario. In the former, a regression model is used to predict the WER score of each single hypothesis that is created through a single automatic transcription channel. In the latter, a ranking model is used to predict the order of multiple hypotheses with respect to their quality. Multiple hypotheses are mainly generated by several ASR systems or several recording microphones.

Application. We introduce two applications in which ASR QE makes salient improvement in terms of WER: *i)* QE-informed data selection for acoustic model adaptation; *ii)* QE-informed system combination. In the

[1]GMM-HMM stands for acoustic models based on hidden Markov models (HMM) with Gaussian mixture models (GMM) to compute the emission probabilities (see §2.3.1)

[2]DNN-HMM stands for acoustic models based on hidden Markov models (HMM) with deep neural networks (DNN) to compute the emission probabilities (see §2.3.2)

former, we exploit single hypothesis ASR QE methods in order to select the best adaptation data for upgrading the acoustic model. In the latter, we exploit multiple hypotheses ASR QE methods to rank and combine the automatic transcriptions in a supervised manner. The experiments on CHiME-3 task, which is consisting of Wall Street Journal utterances recorded by multiple distant microphones in noisy environments, show that QE-informed acoustic model adaptation and QE-informed system combination, respectively, yield 1.8% and 1.7% absolute WER reduction.

The outcomes of this thesis are packed in the frame of an open source toolkit named TranscRater[3] (transcription rating toolkit) which has been developed based on the aforementioned studies. TranscRater can be used to extract informative features, train the QE models and predict the quality of the reference-less recognitions in a variety of ASR tasks.

1.1 Thesis summary

Chapter 2 depicts the architecture of an ASR system. The state-of-the-art acoustic modeling and language modeling methods are described in more detail.

Chapter 3 forms the main body of this thesis and it focuses on ASR QE architecture. This chapter discusses the motivation, history and methods to construct an efficient and applicable ASR QE system. Starting from the first proposal by Negri et al. (2014), who explored ASR QE as a sentence-level WER prediction, to the further extensions by Zamani et al. (2015) who addressed it as binary classification algorithm and de Souza et al. (2015) who tackled the problem of multi domain QE challenge. We categorized these mentioned works as single hypothesis ASR QE, because all of them concentrate on the situation in which only one automatic transcrip-

[3]https://github.com/hlt-mt/TranscRater

tion channel exists. ASR QE in multiple hypotheses scenario is addressed in Chapter 3. For the first time, machine-learned ranking algorithms are applied to an ASR task for ranking the multiple hypotheses according to their recognition quality. At the end of this chapter, the efficacy of the proposed methods for both single hypothesis and multiple hypotheses scenarios is assessed through several experiments.

Chapter 4 describes a novel application of single hypothesis ASR QE to improve unsupervised acoustic model adaptation. The method is based on applying pre-trained QE models to the output of the first decoding pass and use the QE results to inform the adaptation algorithm with the quality of adaptation data. The experiments show that self acoustic model adaptation using automatic transcription of test set becomes more effective, if the part of test data with low quality is removed from the adaptation set. The results confirm that single hypothesis ASR QE methods, described in Chapter 3, can provide useful information about the quality of the adaptation set, and consequently, it helps in improving the accuracy of the new acoustic models.

Chapter 5 explores the application of multiple hypotheses ASR QE to system combination. The preliminary observations show that ROVER (Fiscus, 1997), a popular ASR system combination algorithm, is sensitive to the order of the input hypotheses. We observe that multiple hypotheses ASR QE methods, described in Chapter 3, is capable to provide this order, and consequently, it improves the ROVER's output. The proposed method, named segment-level QE-informed ROVER leads salient WER reduction in two different tasks: combination of multiple ASR systems (IWSLT) and combination of multiple distant microphones (CHiME-3).

1.2 Contributions

At the beginning of this PhD program, we concentrated on neural network language models as the single source of knowledge to estimate the quality of the recognized hypothesis in n-best lists and word graphs. Our efforts in this trend led to minor contributions as listed below:

1. n-best list rescoring using recurrent neural network language model (RNNLM) (Jalalvand, 2013).

2. using minimum error rate training (MERT) to tune the interpolation weights of the acoustic and language model components for performing iterative confusion network decoding by means of long-span RNNLM (Jalalvand and Falavigna, 2013).

3. A* search stack rescoring using RNNLM. Simple n-best list rescoring provides a limited opportunity to exploit RNNLMs. On the other hand, word graph rescoring using long-span LMs is quite expensive in terms of memory and time. In this work, we proposed to exploit RNNLM for rescoring the partial hypotheses inside the A* search stack (Jalalvand and Falavigna, 2014).

4. As the first steps towards ASR QE, we tried to detect the erroneous words using stacked auto encoder based neural network as the classifier. In this work, we exploited suitable word-level features and efficient classifiers to identify the errors and to predict the approximate sentence-level WERs (Jalalvand and Falavigna, 2015).

For the sake of brevity, we avoided to describe the above mentioned methods and results in the main body of this dissertation. The readers are welcomed to read the cited papers.

The major contributions of this thesis with regard to ASR QE are:

1. A comprehensive study on ASR QE including: definitions, features, learning algorithms and applications.

2. Improving the acoustic model adaptation techniques by using ASR QE for consciously selecting adaptation data with high quality (Jalalvand et al., 2015a). In this method, after the first pass of decoding, we estimate the quality of the transcribed data and then, we filter-out those with low quality before performing DNN-HMM adaptation (Falavigna et al., 2016).

3. Exploring ASR QE in multiple hypotheses scenario. For the first time, machine-learned ranking (MLR) algorithms have been used in ASR to predict the quality of hypotheses in a competitive manner (Jalalvand et al., 2015b).

4. Improving ROVER, a well-studied ASR system combination method, by first increasing the granularity of the inputs from utterance (several minutes) to segments (several seconds) and then by ordering the candidates at segment-level using ASR QE approaches (Jalalvand et al., 2015b).

5. Development of TranscRater, an open source toolkit for rating the automatic transcriptions. The toolkit is equipped with all the models and algorithms starting from feature extraction and training to parameter tuning and quality prediction for both single hypothesis and multiple hypotheses scenarios (Jalalvand et al., 2016).

1.3 Publications

The achievements of this PhD have been already published in the top-tier ASR and NLP journals, conferences and workshops.

1.3.1 Journals

- Shahab Jalalvand, Matteo Negri, Daniele Falavigna, Marco Matassoni, Marco Turchi. Automatic Quality Estimation for ASR System Combination. Waiting for review in Computer Speech & Language. 2016.

- Daniele Falavigna, Marco Matassoni, Shahab Jalalvand, Matteo Negri, and Marco Turchi. DNN adaptation by automatic quality estimation of ASR hypotheses. In: Computer Speech & Language. 2016.

1.3.2 Conferences and workshops

- Shahab Jalalvand, Matteo Negri, Marco Turchi, Jos GC de Souza, Daniele Falavigna, and Mohammed RH Qwaider. Transcrater: a tool for automatic speech recognition quality estimation. ACL 2016, pages 43–48, 2016.

- Shahab Jalalvand, Matteo Negri, Daniele Falavigna, and Marco Turchi. Driving rover with segment-based asr quality estimation. In Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL), pages 1095–1105, Beijing, China, 2015b.

- Shahab Jalalvand and Daniele Falavigna. Stacked Auto-Encoder for ASR Error Detection and Word Error Rate Prediction. In Proc. of the 16th Annual Conference of the International Speech Communication Association (INTERPSEECH), pages 2142–2146, Dresden, Germany, 2015.

- Shahab Jalalvand, Daniele Falavigna, Marco Matassoni, Piergiorgio Svaizer, and Maurizio Omologo. Boosted Acoustic Model Learning and Hypotheses Rescoring on the CHiME-3 Task. In Proc. of the

IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pages 409–415, Scottsdale, Arizona, USA, 2015a.

- Shahab Jalalvand and Daniele Falavigna. Direct Word Graph Rescoring Using A* Search and RNNLM. In Proc. of the 15th Annual Conference of the International Speech Communication Association (INTERSPEECH), pages 2630–2634, Singapore, 2014.

- Bagher BaBaali, Romain Serizel, Shahab Jalalvand, Daniele Falavigna, Roberto Gretter, and Diego Giuliani. FBK@ IWSLT 2014-ASR track. In Proc. of the International Workshop on Spoken Language Translation. Lake Tahoe, California, USA, pages 18–25. 2014.

- Shahab Jalalvand and Daniele Falavigna. Parameter Optimization for Iterative Confusion Network Decoding in Weather-Domain Speech Recognition. In Proc. of the 10th International Workshop on Spoken Language Translation (IWSLT), pages 333–337, Heidelberg, Germany, 2013.

- Shahab Jalalvand. Improving language model adaptation using automatic data selection and neural network. In Recent Advances in Natural Language Processing, pages 86–92, Hissar, Bulgaria, September 2013.

- Alexei V Ivanov, Shahab Jalalvand, Roberto Gretter, and Daniele Falavigna. Phonetic and anthropometric conditioning of MSA-KST cognitive impairment characterization system. In Proc. of Automatic Speech Recognition and Understanding (ASRU). Olomouc, Czech Republic, pages 228–233, 2013.

Chapter 2

Automatic Speech Recognition

This chapter describes the architecture of an automatic speech recognition (ASR) system. The modules that will be concentrated in the following chapters are described in more detail. In particular, *output post-processing* module will be focused in Chapter 3, where quality estimation algorithms for ASR outputs form the main body of this thesis; *acoustic modeling* module will be discussed in Chapter 4, where we will propose adaptation methods equipped with automatic quality estimation; and finally *ASR system combination*, another post-processing approach, will be concentrated in Chapter 5, where we will apply automatic quality estimation to ASR system combination.

2.1 Introduction

Automatic speech recognition brings together three knowledge sources:

- linguistics,
- electrical engineering and
- computer science.

Linguistic information is derived from language specific knowledge, mainly

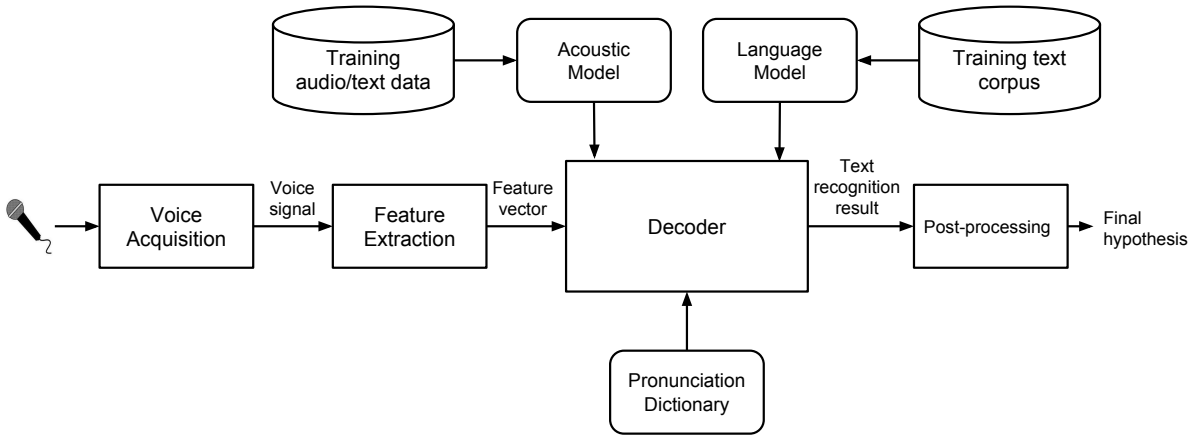


Figure 2.1: ASR architecture

related to phonetics, lexicon and syntax. From electronic engineering, topics such as signal processing, signal enhancement, noise/echo cancellation and acoustic analysis are required. Acoustic and language modeling, as well as search strategies over finite state networks, require competences from computer science.

Figure 2.1 shows the main modules of an ASR system. The speech is first recorded by suitable microphones and digitized. Then the signal is windowed into short frames ($\approx$ 20ms) and from each frame, proper acoustic features are extracted (§2.2). The decoder accepts the sequence of feature vectors and it uses the pre-trained acoustic (§2.3) and language models (§2.4) and the pronunciation dictionary to find the most probable sequence of words (§2.5). The acoustic model is trained on a audio/text corpus and the language model is trained on a text corpus for each specific language. The output of the decoder is usually purified by post-processing approaches such as n-best list rescoring and syntax sanity check and then the final hypothesis is provided.

An ASR system aims to find the most probable sequence of words, $\hat{W}$, generating a sequence of acoustic observations, X:

$$\hat{W} = \underset{W}{argmax}\{P[W|X]\} = \underset{W}{argmax}\{P[X|W] \times P[W]\} \qquad (2.1)$$

$X = x_1, x_2, ..., x_T$ indicates the sequence of acoustic observations, each represented by an acoustic feature vector (§2.2); $P[X|W]$ is the acoustic model likelihood (§2.3) and $P[W]$ is the language model likelihood, i.e. the a priori probability of the sequence of words, W (§2.4). Speech recognition or decoding is the procedure for solving equation 2.1 (§2.5).

2.2 Acoustic feature extraction

The first step of speech recognition is to extract the acoustic features $X = x_1, x_2, ..., x_T$ from speech signal. To this purpose, the speech signal is windowed into short frames (≈20ms) in which it is assumed to be stationary. A time overlap (≈10ms) between subsequent windows is also applied. Common approaches for extracting acoustic features from each frame are based on:

- filter-bank analysis usually spaced according to a perceptual auditory scale (MEL) (Davis and Mermelstein, 1980) and
- linear prediction coefficients (LPC) analysis (Linde et al., 1980).

In both cases, the cepstrum of resulting coefficients is computed via discrete cosine transform (DCT) that respectively yield to Mel frequency cepstral coefficients (MFCCs) and linear prediction cepstral coefficients (LPCCs). A combination of both above-mentioned features gives rise to perceptual linear prediction (PLPs) (Hermansky, 1990). Usually first and second order derivatives (Furui, 1981) are also added to static features (MFCCs, LPCCs or PLPs) in order to form the observation vector for each frame.

Other types of features such as speech loudness, pitch, signal-to-noise ratio are useful in other tasks like speaker identification, accent identification

and emotion detection (Goldwater et al., 2010; Pellegrini and Trancoso, 2010).

2.3 Acoustic Modeling

An acoustic model computes $P[X|W]$, the probability of generating an acoustic observation sequence given a word (or phone) sequence. Usually, the words are represented by their corresponding sequence of phones. Each phone is in turn represented by a left-to-right hidden Markov model (HMM) (Rabiner and Juang, 1986). An HMM is a Markov random process that emits a symbol at each time instance (i.e. at each frame), associated with a probability that depends on the current state and then moves to the next state with a probability that again depends on the current state (Jelinek, 1997). HMM states cannot be directly observed, instead, these are the acoustic features that are observed and this is the reason for using the term "hidden". To compute the probability for a sequence of acoustic observations given a sequence of states, an output observation distribution is assigned to each HMM state $s = j \in [1, J]$. These distributions are traditionally modelled by means of Gaussian mixture models (GMM) or more recently by means of deep neural networks (DNN).

2.3.1 Gaussian Mixture Model HMM

GMM computes the likelihood of the observed acoustic vector x_t at time t in state j by:

$$p[x_t|s = j] = \sum_{m=1}^{M} c_{jm} \mathcal{N}(x_t; \mu_{jm}, U_{jm}) \tag{2.2}$$

where, $\mathcal{N}$ indicates a Gaussian distribution with μ_{jm} and U_{jm} being respectively the mean vector and covariance matrix of the m-th mixture component of state j; c_{jm} ($c_{jm} \geq 0, \sum_{m=1}^{M} c_{jm} = 1$) is the weight of this

component and M is the total number of Gaussian mixture components in state j (De Mori and Brugnara, 1996).

2.3.2 Deep Neural Network HMM

Recently, instead of GMMs, deep neural networks are used to compute the aforementioned likelihood. (Hinton et al., 2012).

In DNN-HMM acoustic models, the state emission probabilities are computed in the output layer of a DNN. This network accepts the input acoustic vector (usually an acoustic counter-vector formed by the concatenation of several frames ($v^0 =< ..., x_{i-1}, x_i, x_{i+1}, ... >$) and passes it through many layers of non-linear transformations. Each neuron i at layer l processes the input vector v^l and computes h_i^l:

$$h_i^l = \sigma(z_i^l(\overrightarrow{v^l})) = \sigma(\overrightarrow{(w_i^l)^T}.\overrightarrow{v^l} + a_i^l) \tag{2.3}$$

In this formula, w^l and a^l are respectively weight matrix and bias, associated to the l-th hidden layer. The input vector v^l is indeed the output of the previous layer, $v^l = h^{l-1}$. Also $\sigma(x) = 1/(1 + exp(-x))$ is the sigmoid function applied element-wise.

The state posterior probability $p[s = j|x_t]$, being x_t an acoustic observation vector at time t, is converted into a state emission likelihood using the following Bayes formula:

$$p[x_t|s = j] = p[s = j|x_t]\frac{p[x_t]}{p[s = j]} \quad 1 \leq j \leq J \tag{2.4}$$

where J is the total number of HMM states and $p[x_t]$ is discarded since it does not depend on the state. $p[s = j|x_t]$ is computed in the output layer L:

$$p[s=j|x_t] = p[s=j|v^L] = \frac{exp(\overrightarrow{(w_j^L)^T}.\overrightarrow{v^L}) + a_j^L}{\sum_{\acute{s}=1}^{J} exp(\overrightarrow{(w_{\acute{s}}^L)^T}.\overrightarrow{v^L} + a_{\acute{s}}^L)} \tag{2.5}$$

where, v^L is the output of the last hidden layer; w^L and a^L are respectively the weight matrix and bias of the output layer; $\acute{s} \in [1, J]$ ranges over all the output neurons that are indeed the representatives of the HMM states.

For training, a possible criterion for estimating weights and biases of the DNN is to minimize the negative cross-entropy $\mathcal{C}(\hat{p}, p)$ between a target distribution $\hat{p}$ and the estimated one over training utterances:

$$\mathcal{C}(\hat{p}, p) = \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{J} \hat{p}[s=j|x_t] \log p[s=j|x_t] \tag{2.6}$$

where T is the total number of frames in the training utterances. Usually, the entries $\hat{p}[s=j|x_t]$ in the target distribution are obtained by forced alignment using an existing GMM-HMM-based ASR system and assuming the value of 1 for the aligned states and zero for all the others.

2.4 Language Modeling

Language model (LM) predicts the probability of a word, given a context of previously observed ones. Among many approaches to build an LM, two widely-used ones are based on n-gram statistics and neural networks.

2.4.1 n-gram Language Model

An n-gram LM is based on n-gram statistics and it computes the conditional probability of seeing a word, given the $n-1$ previous words (Chen and Goodman, 1999). The probability of a sequence of words $P[W]$ is computed as follows:

$$P[W] = P[w_1, ..., w_N] \approx \prod_{i=1}^{N} p(w_i | w_{i-(n-1)}^{i-1}) \tag{2.7}$$

where, the n-gram probabilities are estimated through their corresponding counts in a training text corpus:

$$p(w_i | w_{i-n+1}^{i-1}) = \frac{count(w_{i-n+1}^{i})}{count(w_{i-n+1}^{i-1})} \tag{2.8}$$

One issue with Equation 2.7 is data sparseness, that is, many of the n-grams are rare and their probability is very low or sometimes zero. To overcome this, Katz (1987) proposed a back-off model. In this model, if one n-gram is rare, its conditional probability will be estimated by the back-off probability of the shorter context.

$$p_{bo}(w_i | w_{i-n+1}^{i-1}) = \begin{cases} d_{w_{i-(n-1)} \dots w_i} \frac{count(w_{i-(n-1)}^{i})}{count(w_{i-(n-1)}^{i-1})}, & if count(w_{i-(n-1)}^{i}) > k \\ \alpha_{w_{i-(n-1)} \dots w_{i-1}} P_{bo}(w_i | w_{i-(n-2)}^{i-1}), & otherwise \end{cases} \tag{2.9}$$

In this formula, $p_{bo}(w_i | w_{i-n+1}^{i-1})$ is the back-off probability of observing w_i; $count(W)$ is the frequency of the sequence W; k is a threshold for the least acceptable number of appearances and d is the Good Turing discounting estimation. Other extensions such as modified Kneser-Ney (Chen and Goodman, 1999) also called modified shift-beta smoothing is widely considered the most effective method of smoothing due to its use of absolute discounting by subtracting a fixed value from the probability's lower order terms to omit n-grams with lower frequencies.

2.4.2 Neural Network Language Model

Neural network LM (NNLM) has several capabilities compared to n-gram model. NNLM alleviates the problem of unseen word sequences by learning

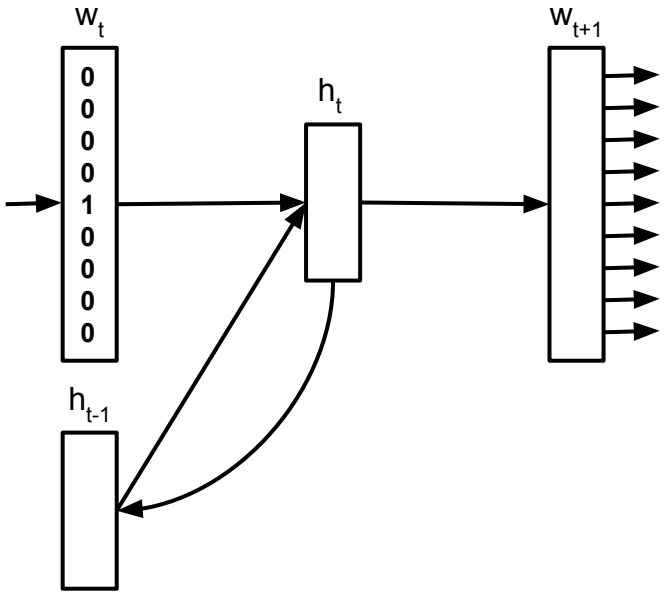


Figure 2.2: RNNLM structure

a distributed representation for each word (word feature vector) along with a probability function for word sequences (Bengio et al., 2003). Continuous-space NNLM is able to map the words from discrete space into continuous space thanks to its projection layer (Schwenk, 2013) that converts the one-hot vectors into compressed vectors.

In particular, the usage of recurrent neural networks (RNN) has shown significant performance in comparison to n-gram LMs and other types of feed forward NNLMs. This is mainly due to the capability of RNNLMs to consider longer contexts than $n - 1$ (Mikolov et al., 2010). Furthermore, the interpolation capability of RNN allows mitigating the effects of unseen word sequences in the training set. In addition, unlike n-gram LM that enlarges exponentially by increasing the size of the training data, the size of RNNLM only depends on the predefined parameters (i.e. the number of hidden layers and the number of hidden neurons).

The structure of an RNNLM is depicted in Figure 2.2. In the input layer,

the words are represented by one-hot vectors, i.e. by a vector with equal dimension to the vocabulary size in which only the index of the desired word is one and the others are zero. The output layer has the same size as the input layer. The application of a softmax function to the output units ensures that each value is between 0 and 1 and the sum of all the values is 1. Therefore, the i-th value in the output layer can be interpreted as the probability of word w_i.

Training procedure starts by randomly initializing the weights. By observing one word in the training set, the probability of the next word is computed using the initial weights. An objective function computes the cost of generating the correct word in the output layer. The cost (error) is back-propagated through the network to update the weights. For RNNLM, a specific type of training named back-propagation through time (BPTT) is used (Boden, 2002).

In spite of the advantages of RNNLM, it is difficult to use this model in the ASR decoder at the first decoding step. Because RNNLM, differently from n-gram LM, does not account explicitly for back-off transitions. This enlarges the search space exponentially. A method for mapping an RNNLM into a back-off LM has been proposed in (Liu et al., 2014). Another method for conversion of RNNLM to weighted finite state transducers (WFST) is proposed in (Lecorvè et al., 2012). Nevertheless, the simplest and widely-used way to take advantage of long span capability of RNNLMs is re-scoring of n-best lists. However, the search space of the n-best list is limited to n.

As an alternative method, we propose to perform *A* stack rescoring* instead of n-best list or word graph rescoring (Jalalvand and Falavigna, 2014). This method rescores the partial hypotheses inside the A* stack. Whenever a new node is expanded and the new hypotheses are pushed into the stack, the partial hypotheses inside the stack are rescored and reordered by means of RNNLM. Therefore, the partial hypothesis with the

highest score (resulted by interpolation of graph score and RNNLM score) is positioned on the top of the stack. Correspondingly, in the next step, this partial hypothesis will be selected as the first node to expand the path.

2.5 Decoding

In large vocabulary continuous speech recognition (LVCSR) systems, the words are usually represented by their corresponding phones. In this way, only a limited set of phonetic models are needed to be trained for each language. Therefore, Equation 2.1 is modified to 2.10 in order to include the phone units for each word sequence:

$$\hat{W} = \underset{W}{argmax} \ \{P[X|u(W)] \times P[W]\} \tag{2.10}$$

In this equation, $u(W)$ represents either a sequence of phones or a sequence of context-dependent phonetic unites (like triphones); $P[X|u(W)]$ is the acoustic likelihood and $P[W]$ is the language model likelihood.

In Equation 2.10, $\hat{W}$ is selected among the possible solutions in a search space. Such space can be represented by a finite state automata (FSA) whose arcs are assigned to n-grams. Usually the starting arc of each n-gram has associated the corresponding LM probability[1]. The "null" transitions (i.e. transitions that do not emit any symbol) are also added to account for back-off probabilities. Once the FSA is built, the lexical model is integrated by means of basic sub-graph substitution. In this way, each arc (i.e. each n-gram) in the FSA will be replaced by a sub-graph corresponding to its phone pronunciation. In the same way each phone (or triphone) in the FSA is replaced by its corresponding HMM model forming a finite state network (FSN).

[1]This early application of LM probabilities allows achieving a more effective pruning during the forward search.

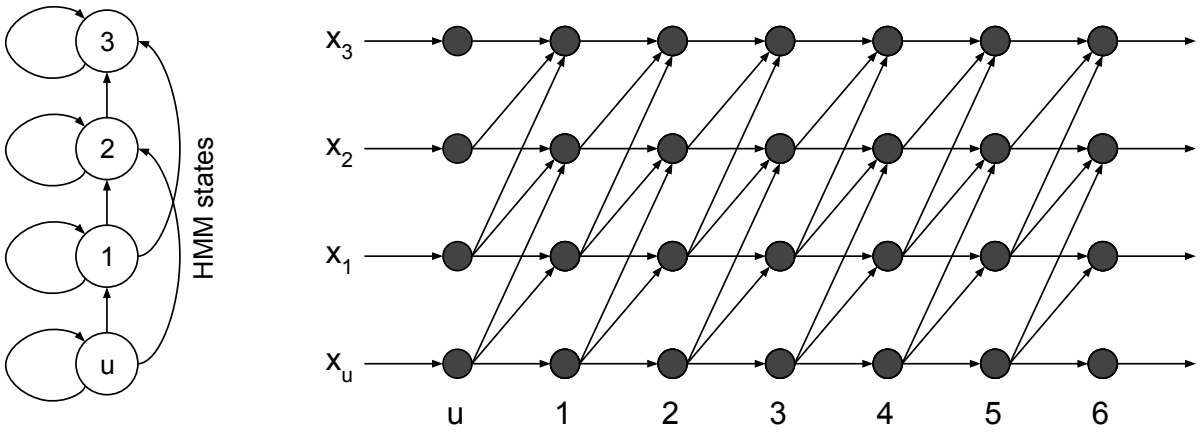


Figure 2.3: A state-time trellis.

The FSN is searched along a data structure called "trellis" (Rabiner and Juang, 1993), depicted in Figure 2.3. Every trellis column holds the values of one of the probabilities for a partial sequence ending at different time instants, and every interval between two columns corresponds to an input frame. The arrows in the trellis represent model transitions, composing possible paths from the initial time instant to the final one. The computation proceeds in a column-wise manner. At every time frame, the scores of the nodes in a column are updated by means of recursion formulas which involve the values of an adjacent column, the transition probabilities of the models, and the values of the output distributions for the corresponding frame (De Mori and Brugnara, 1996).

2.5.1 Word Graph generation

A word graph (WG) is an acyclic graph whose transitions are associated to the words in the recognition dictionary. Each transition contains information about: starting and ending time of each transition word and its corresponding acoustic and language model scores. A WG contains the information about word-endings as they occur in the course of the left to right decoding pass (Aubert et al., 1994). The approach proposed in

(Aubert et al., 1994) takes full advantage of the bigram LM to constrain the graph, without requiring any further optimization or pruning stage. More precisely this method relies on the assumption that the position of a word depends only on the word pair under consideration.

WG is also used to compute the word posterior probabilities. This computation is performed in two steps. First the well known forward-backward algorithm is used to calculate a link posterior probability for each link in the graph. The link posterior $p(l|X)$ is defined as the sum of the probabilities of all paths q, passing through the link l, normalised by the probability of the signal $p(X)$:

$$p(l|X) = \frac{\sum_{Q_l} p(q, X)}{p(X)} \tag{2.11}$$

where $p(X)$ is approximated by the sum over all paths through the lattice. The probability of a path $p(q, X)$ is composed by the acoustic likelihood $p_{acc}(X|q)$ and the language model probability $p_{lm}(W)$:

$$p(q, X) = p_{acc}(X|q)^{1/\gamma} p_{lm}(W) \tag{2.12}$$

γ is used to scale down the acoustic likelihood.

Each recognized word may include a large number of links. Therefore, in order to obtain the word posterior, the link posteriors need to be combined, as explained below.

2.5.2 Confidence Measure computation

The confidence score or confidence measure indicates how certain is the ASR decoder about each word hypothesis. Word posterior probabilities, provided in the word graph as described in the previous subsection, can be used directly as confidence measure for the word hypothesis. Three different ways are proposed in (Wessel et al., 2001) to calculate the confidence

measure of word w in link $l]_s^e$:

1. sum over all the links which are labelled by w and intersect with the time boundary $[s, e]$;

2. sum over all the links which are labelled by w and intersect with the median time frame of link l;

3. take the maximum from the two mentioned methods. That is, once we sum over all the links, labelled by w which intersect with $[s, e]$ and then we sum over all the links with label w which intersect the median time frame of the link l and finally we take the maximum as the confidence measure for the word w in link l.

The confidence measure for a transcribed sentence can be easily achieved by computing the average of its word-level confidence measures. Such score, however, tends to overestimate the word confidences. From this point of view, confidence measure is not a proper metric to automatically supervise the performance of different ASR systems, because each system may generate high confidence scores for its hypotheses. Another drawback of confidence measure is its dependency on the ASR decoder. That is, in a black-box condition, when the inner behavior of the ASR system is not known, we cannot compute confidence scores. This thesis addresses these problems and it aims to evaluate the quality of the ASR output in a more reliable manner, without access to the human-craft references.

Chapter 3

ASR Quality Estimation

In the previous chapter, an ASR architecture was described from signal acquisition to output production. This chapter explores ASR quality estimation (QE) by discussing the motivations, architectures and methods to construct an efficient ASR QE system. The successive chapters (4 and 5) will introduce two applications for ASR QE that lead to significant WER reduction.

3.1 Introduction

In a natural language processing (NLP) system, the quality of the output can be addressed from different aspects. For instance in machine translation, three aspects for the translation quality can be considered: fluency, adequacy and complexity (Camargo de Souza, 2016). These factors can be extended to automatic speech recognition as well. Therefore, a recognition is:

- *fluent* when it conforms the syntax and grammar of the desired language;
- *adequate* when it conveys the same meaning as its source and

- *complex* when the source signal is difficult to be recognized, because of noise, echo, reverberation, accent, etc.

The evaluation methods for NLP systems can be categorized into four groups, based on availability of information resources:

Manual reference. When the manual reference is available, system performance can be evaluated simply by aligning the output with the reference. This alignment is usually measured by a metric such as word error rate (WER) [1] (Jelinek, 1997) in ASR, translation error rate (TER) [2] (Snover et al., 2006) and bilingual evaluation understudy (BLEU) [3] (Papineni et al., 2002) in machine translation.

Confidence measure (CM). When there is no manual reference available, but the decoding information (such as word graphs) is available, then CM can indicate, to some extent, the quality of the output (§2.5.2). This measure is a function of input, output and the decoder information and it is usually between 0 and 1. CM shows how confident is the NLP system about the generated output.

Confidence estimation (CE). When there is no manual reference available, but the decoding information is available, and in addition, an external source of knowledge such as part-of-speech tag is accessible, then the confidence measure can be estimated by an external model (Gandrabur et al., 2006).

Quality estimation (QE). When neither the manual reference nor the decoding information is available, then the quality of the NLP system can

[1]The word error rate (WER) is the minimum edit distance between an hypothesis and the reference transcription. Edit distance is calculated as the number of edits (word insertions, deletions, substitutions) divided by the number of words in the reference.

[2]TER computes the minimum amount of edit operations (insertions, deletions, substitutions and shifts of words) required to transform the translated segment into the reference segment divided by the average number of words in the reference(s)

[3]BLEU is a simple metric that matches different n-gram sizes between the MT output and one or more references.

be estimated using an external model with features extracted from inputs and outputs. QE focuses on estimating absolute measures of quality without access to the internal knowledge sources. This topic has been widely studied in many NLP tasks. For example, in machine translation (MT), QE with the goal of bypassing the need of human-created reference translations has motivated a large body of research. The motivations (cost effective quality prediction at run-time) and the methods (supervised learning, either as regression or multi-class classification) are, indeed, the same in all NLP tasks. For a complete overview of the current approaches to MT QE, the reader is referred to the comprehensive overviews published within the yearly Workshops on Statistical Machine Translation (Callison-Burch et al., 2012; de Souza et al., 2013, 2014; Bojar et al., 2015, 2016) and to the works dealing with quality prediction at word level (Ueffing and Ney, 2007; Bach et al., 2011), sentence level (Specia et al., 2009) and document level (Soricut and Echihabi, 2010).

Napoles et al. (2016) investigates the utility of grammatically-based, reference-less QE metrics for evaluation of grammatical error correction (GEC) systems. GEC systems strongly rely on manual references. The authors show that these QE metrics correlate very strongly with human judgments and they are competitive with the leading reference-based evaluation metrics.

Another field, in which QE has been successfully used is automatic speech recognition (Negri et al., 2014; Ng et al., 2015a,b). In spite of years of study on ASR evaluation metrics, confidence measure and confidence estimation, ASR QE has not been investigated sufficiently. Negri et al. (2014) introduced ASR QE as a WER prediction problem. They analyse different learning algorithms and features in various testing conditions. In the successive sections, we focus on ASR QE and we review the recent progresses.

The current chapter, as the core of this thesis, describes the ASR QE problem, the features and the learning algorithms. In §3.2, we describe the structure of an ASR QE system. §3.3 introduces the features suitable for training the QE models. Then, we explain the learning algorithms for two different scenarios. The first one is dedicated to QE for single hypothesis and it mainly discusses about word error rate prediction at sentence-level (§3.4). The second one describes QE for multiple hypotheses and it refers to the condition in which there are several transcription channels, e.g. generated by different ASR systems or several recording microphones (§3.5). The performance of the proposed ASR QE system is evaluated through several experiments in §3.7.

3.2 ASR Quality Estimation (ASR QE)

Everyday, million hours of speech data including TV programs, YouTube videos, meetings, telephone conversations are automatically transcribed into text by ASR systems. A large amount of these texts cannot be rated because:

- the manual references are not available;
- hand-craft evaluation is too costly in terms human source;
- although one can approximately assess the quality of the ASR output through the confidence measure (§2.5.2), in many ASR applications, especially the commercial ones the confidence measure and the inner information of the recognizer are not provided[4].

The above mentioned issues motivate the need for an automatic ASR evaluation algorithm that is reference-free and confidence-independent.

[4]Consider, for instance, the explosion of captioned YouTube videos available on the Web. As announced by Google, in 2012 about 157 million YouTube videos in 10 languages already featured captions generated by a black-box ASR system (source: http://goo.gl/5Wlkjl).

Negri et al. (2014) propose a supervised regression algorithm to predict the WER of automatically transcribed audio recordings. The authors analyse the capability of different learning algorithms with different features to predict sentence-level WER scores. They show that even in difficult testing conditions where the training and test data are from different domains, and also the ASR decoder information is not accessible, they can closely approximate the true WER, by using suitable regression models, namely extremely randomized tree (XRT) (Geurts et al., 2006) and appropriate feature sets that are described in §3.3.

de Souza et al. (2015) explores ASR QE by focusing on the problem of domain mismatches between training and test data. Indeed, as pointed out by (Negri et al., 2014), simple supervised learning methods are very sensitive to large variations in the distribution of the instances in the two sets (both at the level of labels and at the level of features). The proposed solution relies on multitask learning to train robust models that exploit the similarities and differences between possibly related tasks, transferring knowledge across them. Results show that the approach is able to take advantage of data coming from heterogeneous domains and it significantly improves over single-task learning baselines, both in regression and in classification. These findings suggest the reliability of ASR QE in particularly challenging test conditions.

The contributions of this thesis make significant extensions on ASR QE by:

- adding word-level features inspired by error detection tasks (Goldwater et al., 2010; Tam et al., 2014);
- proposing machine-learned ranking (MLR) algorithms (Hang, 2011; Clemencon et al., 2013) for multiple hypotheses scenario;
- introducing novel applications of ASR QE to improve WER results.

Figure 3.1 shows the architecture of an ASR QE system, starting from feature extraction to machine learning modules. The features are extracted from both signal and automatic transcription at sentence-level. These features along with the corresponding true WER scores (as target values) are used to train the QE models. The models are then applied to test feature vectors to predict their WER scores. In the successive sections, we describe these modules in more detail.

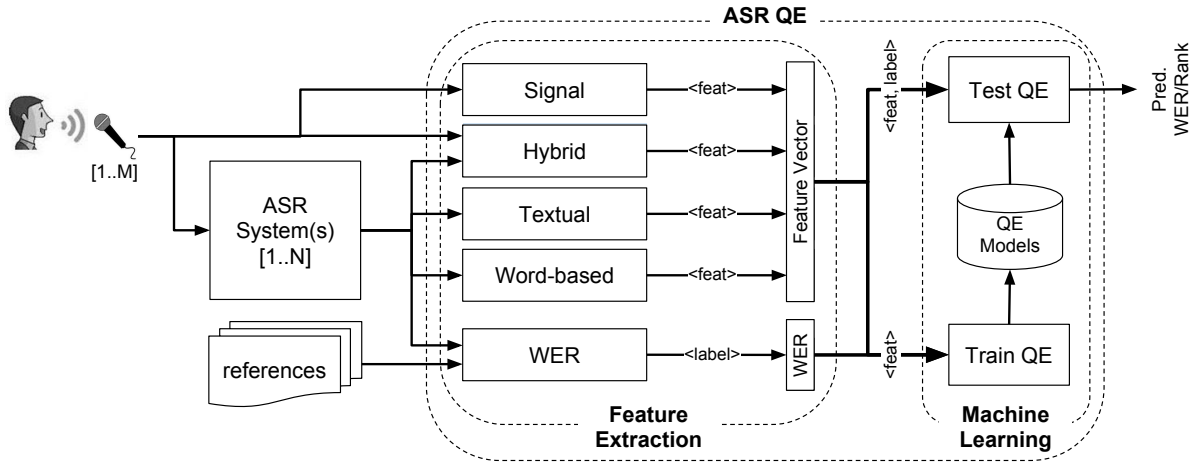


Figure 3.1: The overall architecture of ASR QE.

3.3 Feature extraction

As mentioned before, one of our assumptions is that the inner information of the ASR decoder is not available. Therefore, the only inputs to the feature extraction module are speech signals and their corresponding automatic transcriptions. Given these inputs, the module computes the features that represent the quality of the transcriptions. These features can be grouped into four major categories: signal, textual, hybrid and word-level.

- **Signal** features aim to capture the difficulty of transcribing a given

speech signal by looking at the signal as a whole. They are obtained by analyzing the audio waveform with a window of 20ms length and overlap of 10ms.

- **Hybrid** features provide a more fine-grained representation of the difficulty of transcribing the signal by considering the time boundary of the recognized words.

- **Textual** features aim to capture the plausibility (*i.e.* the fluency) of a transcription.

- **Word-based** features aim to capture word pronunciation and recognition difficulty. For each recognized word, these features are obtained by counting the number of homophones/lexical-neighbors and by computing the word-level language model probabilities. For the former, we use a pronunciation dictionary and for the latter we use a set of n-gram and neural network language models (Mikolov et al., 2010). Our preliminary experiments showed that using a set of language models trained on both in-domain and out-of-domain corpora has positive influence in learning process (Jalalvand et al., 2015b).

A complete list of these features is reported in Table 3.1. The following sections describe the machine learning module by taking two scenarios into account: single hypothesis and multiple hypotheses.

3.4 Machine learning: Single hypothesis ASR QE

For this scenario, we consider two learning algorithms: regression and classification. The former aim to predict the WER of each hypothesis and the latter aim to identify good/bad transcriptions.

Signal (17)	mean values of 12 Mel Frequency Cepstral Coefficients (MFCCs) removing the 0^{th} order coefficient is discarded (12), log energy computed on the whole segments (1), the mean/min/max values of raw energy (3), total segment duration (1).
Textual (10)	number of words (1), LM log probability (1), LM log probability of part of speech (POS) (1), log perplexity (1), LM log perplexity of POS (1), percentage (%) of numbers (1), % of tokens which do not contain only "[a-z]" (1), % of content words (1), % of nouns (1), % of verbs (1).
Hybrid (26)	signal-to-noise ratio (SNR) (1), mean/min/max noise energy (3), mean/min/max word energy (3), (max word - min noise) energy (1), number of silences (#sil) (1), #sil per second (1), number of words (#wrd) per second (1), $\frac{\#sil}{\#wrd}$ (1), total duration of words (D_{wrd}) (1), total duration of silences (D_{sil}) (1), mean duration of words (1), mean duration of silences (1), $\frac{D_{sil}}{D_{wrd}}$ (1), $D_{wrd} - D_{sil}$ (1), standard deviation (std) of word duration (1), std of silence duration (1), mean/std/min/max of pitch[5] (4), number of hesitations (1), frequency of hesitations (1).
Word (22)	POS-tag/score of the previous/current/next words (6), RNNLM probabilities given by models trained on in-domain/out-of-domain data (2), in-domain/out-of-domain 4-gram LM probability (2), number of phoneme classes including fricatives, liquids, nasals, stops and vowels (5), number of homophones (1), number of lexical neighbors (heteronyms) (1) binary features answering the three questions: "is the current word a stop word?"/"is the current word before/after repetition?"/"is the current word before/after silence?" (5).

Table 3.1: A complete list of 75 features for training ASR QE models.

3.4.1 Regression

This method exploits a supervised regression model to predict the WER at sentence-level. Assume that a baseline ASR system including acoustic model and language model is already trained on a separated training corpora. This ASR system has been used to transcribe the development and test sets. Note that this ASR system is in black-box, that is the provided transcriptions are simply word sequences with the word level time boundaries. For the development set, we also have the manual reference with which we can compute true WER scores at sentence-level. Whereas, for the test set, there is no manual reference available. The task is to predict

the WER of test hypotheses using a regression model that is trained on the the results of development set.

Negri et al. (2014) compares two regression models based on support vector regression (SVR) (Smola and Schölkopf, 2004) and extremely randomized tree (XRT) (Geurts et al., 2006):

- **SVR.** With this model, the goal is to find a function that yields a small deviation from the true WER for all the development data, and at the same time, this function must be as flat as possible (Smola and Schölkopf, 2004).

- **XRT** is a tree-based ensemble method for supervised classification and regression[6]. In XRT, each tree can be parametrized differently. When a tree is built, the node splitting step is done by picking the best split among a random subset of the input features. The results of the individual trees are then combined by averaging their predictions.

The hyper-parameters of both SVR and XRT models are optimized using randomized search (Bergstra and Bengio, 2012). In the experiments, we use both learning methods as implemented in the Scikit-learn package (Pedregosa et al., 2011).

3.4.2 Classification

Instead of assigning continuous numbers (i.e. WERs) as the labels, an alternative approach makes use of explicit good/bad labels. This has the advantage of avoiding the user, the burden of interpreting scores in the [0, 1] interval. In particular, binary quality predictions would help in tasks like: *i)* deciding if an utterance in a dialogue application has been correctly recognized, *ii)* deciding if an automatic transcription is good enough for

[6]XRT has been also successfully used for MT quality estimation (de Souza et al., 2013; C. de Souza et al., 2014).

the corresponding audio recording or if it needs manual revision (e.g. in subtitling applications), *iii)* selecting training data for acoustic modelling based on active learning (Riccardi and Hakkani-Tur, 2005), and *iv)* retrieving audio data with a desired quality for subsequent processing in media monitoring applications.

In (Zamani et al., 2015), two classification strategies are experimented. The first strategy, i.e. classification via regression, represents the easiest way to adapt the method proposed in (Negri et al., 2014). It fits a regression model on the original training instances, applies it to the test data, and finally maps the predicted regression scores into good/bad labels according to a threshold τ. The second strategy, i.e. standard binary classification, can be implemented by labeling the training data into good/bad instances according to τ, training a binary classifier on such data, and finally applying the learned model on the test set.

Both strategies have pros and cons that are worth to consider. On one side, classification via regression directly learns from the WER labels of the training samples. In this way, it can effectively model the instances whose WERs are far from the threshold τ, but at the same time, it is less effective in classifying the instances with WER values close to τ. Moreover, in case of skewed label distributions, its predictions might be biased towards the average of the training labels. Nevertheless, since such mapping is performed a posteriori on the predicted labels, the behaviour of the model can be easily tuned with respect to different user needs by varying the value of τ. On the other side, standard classification learns from binary labels obtained by mapping a priori the WER labels into the two classes. This means that the behaviour of the model cannot be tuned with respect to different user needs once the training phase is concluded (to do this, the classifier should be re-trained from scratch). Also, standard classification is subject to biases induced by skewed label distributions, which typically

results in predicting the majority class. To cope with this issue, Zamani et al. (2015) applies instance weighting (Veropoulos et al., 1999) by assigning to each training instance a weight, computed by dividing the total number of training instances by the number of instances belonging to the class of the given utterance.

Since classification via regression and standard classification are potentially complementary strategies, Zamani et al. (2015) also investigates the possibility of joint contribution of the two strategies. To this aim, a stacking method (Wolpert, 1992) is used that consists in training a meta-classifier on the predictions returned by an ensemble of base classifiers. To do this, training data is divided in two portions. One is used to train the base estimators; the other is used to train the meta-classifier. In the evaluation phase, the base estimators are run on the test set, their predictions are used as the features for the meta-classifier, and its output is returned as the final prediction.

3.5 Machine learning: Multiple hypotheses ASR QE

In many ASR scenarios, there are several automatic transcriptions for a single utterance. These transcriptions can be produced by several ASR engines (Basson et al., 2003; Jalalvand et al., 2015b), by several microphones (Barker et al., 2015; Vincent et al., 2013; Barker et al., 2013) or a mixing of both. For example, in IWSLT2013 workshop[7] that was dedicated to spoken language translation, the ASR submissions from all the participants were combined together, in order to provide the data for the machine translation track. In Chapter 5, it will be shown that ASR QE can improve the combination process, by automatically ordering the input

[7]The International Workshop on Spoken Language Translation (IWSLT – `http://workshop2013.iwslt.org/`) is a yearly workshop associated with evaluation campaign on spoken language translation.

components according to their predicted quality.

Two strategies for ranking multiple hypotheses using ASR QE are investigated in this chapter:

- ranking by regression and
- machine-learned ranking.

3.5.1 Ranking by regression

The predicted WER scores, generated by the regression models, can be used not only as a measure of quality for automatic transcriptions, but also for ranking the multiple hypotheses. Ranking by regression tries to predict the WER of each hypothesis, independently from the others. Experiments show that ranking by regression is able to predict the order of the hypotheses with a strong correlation to the oracle orders (see §3.7.2).

3.5.2 Machine-Learned Ranking

A more effective ranking strategy is based on machine-learned ranking (MLR) algorithms that are widely exploited in information retrieval and question answering tasks (Cao et al., 2007; McFee and Lanckriet, 2010; Clemencon et al., 2013).

MLR performs a pairwise comparison between the candidates (Cao et al., 2007). For each pair of automatic transcriptions, it processes the corresponding feature vectors and decides to place one transcription ahead of the other, returning a score for this decision. Thanks to this score, the algorithm is able to rank more than two candidates.

In the experiments reported in this chapter, we compare two ranking models. The first one, called RANKNET (Burges et al., 2005), makes use of neural networks. The second one makes use of random forest (Jiang, 2011) and it is based on training multiple decision trees and ensemble.

To train the ranking models, the samples are provided as $< features, RANK >$ tuples. $RANK$ labels are computed with respect to the true WER of all hypotheses corresponding to each utterance. The lower WER, the lower $RANK$ value. Note that, in this case the issue of tied ranks arises. This issue is well addressed in the experiments in Chapter 5.

3.6 Evaluation metrics

The hyper-parameters of the prediction models such as the number of trees, the number of leaves per tree and feature selection rate in XRT models must be tuned with regard to appropriate metrics.

For the models in single hypothesis ASR QE, mean absolute error (MAE) between the predicted WERs ($predWER$) and true WERs ($trueWER$) is used. The lower MAE, the closer predicted scores to the true WERs. Given K utterances, MAE is computed by:

$$MAE = \frac{1}{K} \times \sum_{k=1}^{K} |predWER_k - trueWER_k|$$

For the models in multiple hypotheses ASR QE, normalized discounted cumulative gain (NDCG) metric is used. The higher NDCG, the stronger correlation between predicted and true rankings (Järvelin and Kekäläinen, 2002).

$$NDCG = \frac{1}{K} \sum_{k=1}^{K} NDCG_k@L$$

$$\text{where,} \quad NDCG_k@L = \frac{DCG_k@L}{IDCG_k@L}$$

$$\text{where,} \quad DCG_k@L = \sum_{l=1}^{L} \frac{2^{rel_l} - 1}{log_2(l+1)}$$

The $NDCG$ score on the whole dataset is computed by averaging the $NDCG_k@L$ scores of all the utterances. L is the number of transcriptions channels (like the number of ASR systems or the number of microphones). $NDCG_k@L$ is obtained by dividing the $DCG_k@L$ score by the ideal score, $IDCG_k@L$, which is resulted by oracle ranking. To compute $DCG_k@L$, we define $rel_l = L - predRANK$. The reason is that in IR field, rel_l refers to the relevance of the item predicted at l-th position. While in our experiments, the item with the lower predicted rank represents the lower WER, and therefore, it indicates higher relevance to the reference sentence.

3.7 Experiments

The performance of the proposed features and learning algorithms are evaluated on data collected for the 3^{rd} CHiME challenge [8]. This audio database consists of English sentences from Wall Street Journal corpus, uttered by four speakers in four noisy environments: bus, cafe, pedestrian area, and street junction. These utterances are recorded by five microphones placed on the frame of a tablet PC (a sixth one is placed on the back, mainly for recording background noise). Development and test sets contain 1,640 and 1,320 sentences, respectively. Automatic transcriptions are produced by a baseline ASR system, provided by the task organizers, which uses Kaldi toolkit (Povey et al., 2011). For the details of the baseline ASR system, please refer to (Barker et al., 2015). Note that in all the experiments reported here, the recognition is performed with this ASR system and the only available inputs are the speech signals and their automatic transcriptions.

The first goal is to assess ASR QE performance in single hypothesis scenario (i.e. WER prediction), when only one of the microphones (the

[8] `http://spandh.dcs.shef.ac.uk/chime_challenge/chime2015/data.html`

5-th one) is used. The second goal is to assess ASR QE performance in multiple hypotheses scenario (i.e. ranking prediction), when the signals from all 5 microphones are automatically transcribed.

3.7.1 Single hypothesis

In this scenario, we train the regression models (SVR and XRT) on a training set formed by automatic transcriptions of the development utterances (1640 utterances). Three sets of features are used: SIG, TEX and SIG+TEX. SIG contains the signal features, described in Table 3.1. TEX is consisting of all the textual, hybrid and word-level features, again described in Table 3.1. Finally SEG+TEX contains both groups. The models are then used to predict the WERs of the test set formed by 1320 utterances. More detail about CHiME-3 dataset including the individual WER results of each microphone can be found in §4.4.1. The results are compared using MAE measure (Equation 3.6). The lower MAE indicates that the predicted WERs are closer to the true ones.

Model-Feature	SIG	TEX	SIG+TEX
Baseline	32.1		
SVR	22.9	24.7	20.3
XRT	20.8	22.1	**18.3**

Table 3.2: MAE (↓) results using regression models in single hypothesis mode.

Table 3.2 shows the MAE results of different feature groups with both XRT and SVR learning algorithms. The baseline result is obtained by assigning the average WER of the development instances, equally to all the test instances. In terms of MAE, SVR model with SIG features already outperforms the strong baseline. Using TEX features also improves over the baseline, though less than the SIG ones. A salient improvement is achieved by combining the two groups of features, indicating that they

carry complementary information. Finally by replacing SVR with XRT algorithm, MAE reduction of 2% is achieved. This is in-line with the results reported by Negri et al. (2014), where they also conclude that the XRT algorithm outperform SVR for this task.

The best MAE result (18.3%), obtained by XRT models with all available features (SIG+TEX) is indeed 13.8 absolute points better than the considered baseline. This confirms the efficacy of the proposed model and features in case of single hypothesis ASR QE. Performing this last experiment using the TranscRater[9] toolkit on a machine with eight Intel Xeon E3-1270 x3.40GHz processors takes 4':15" time. This time includes 3":30' for extracting all the features from 1640 training and 1320 test utterances and 45" for training and testing the XRT model with 100 iterations. It is worth mentioning that the feature extraction time is highly dependent on the number of signals and the size of the language models for extracting the features. For these experiments, we use four language models: an RNNLM trained on 37MW corpus provided by the CHiME-3 organizers, an RNNLM trained on a 10MW corpus, automatically selected from news data by using the training references as the seeds, a 4-gram LM trained on WSJ corpus and a 4-gram LM trained on the mentioned 37MW corpus.

3.7.2 Multiple hypotheses

In this mode we use all 5 microphones provided in CHiME-3 dataset. The QE performance is measured by the NDCG score (Eq. 3.6). The higher NDCG value, the better ranking performance. The baseline results are computed by averaging the NDCG scores obtained from one hundred iterations of randomly ranked instances.

The results reported in Table 3.3 have been achieved using ranking by regression §(3.5.1). In this method, the regression models are trained on

[9]https://github.com/hlt-mt/TranscRater

Model-Feature	SIG	TEX	SIG+TEX
Baseline	73.6		
SVR	81.0	81.7	84.1
XRT	80.7	83.3	**85.0**

Table 3.3: NDCG (↑) results using regression models in multiple hypotheses mode.

1640*5=8200 samples[10]) from transcriptions of development set and then we apply them to the transcriptions of the test set (1320*5=6600 samples). Afterwards, the hypotheses are ranked according to the predicted values and the NDCG score is computed. As expected, XRT outperforms SVR and it shows the best performance when all the features are used. The large NDCG improvement over the baseline (+11.4) seems to make this combination particularly suitable for ranking by regression.

Model-Feature	SIG	TEX	SIG+TEX
Baseline	73.6		
RANKNET	83.1	85.5	87.5
RF	84.6	86.6	**88.2**

Table 3.4: NDCG (↑) results using ranking models in multiple hypotheses mode.

The results reported in Table 3.4 show the performance of machine-learned ranking approaches (§3.5.2). As the ranking algorithms, RANKNET and random forest (RF) are utilized, both are implemented in (Dang, 2013). RANKNET with SIG features already improves over the baseline (9.5% absolutely better than the baseline). TEX features results 85.5% NDCG score which is 0.5% better than the best result obtained with ranking by regression (Table 3.3). Combination of all features makes further improvement (87.5%), showing once again the complementarity between the SIG and textual features. By replacing RANKNET with RF, we observe consistent improvement in terms of NDCG score. The best result (88.2%), obtained

[10]As we use 5 different microphones that consequently provide 5 different transcriptions per utterance

by RF models with all features (SIG+TEX) is indeed +14.6% absolutely better than the baseline, 3.2% better than ranking by regression and 0.7% better than RANKNET. The main reason that the ranking algorithms are better than ranking by regression is in their capability to perform pairwise comparison.

The next chapters show how ASR QE can helps in reducing WER results in both single and multiple hypotheses scenarios.

3.8 Summary

Automatic speech recognition quality estimation (ASR QE) was studied thoroughly in this chapter. ASR QE was addressed as an alternative for confidence estimation (CE) when the ASR decoder is not known and decoding information such as word graph, N-best lists, acoustic model and language model scores are not available. We described ASR QE from both feature engineering and learning algorithms perspectives. We introduced four sets of features including signal-based, textual-based, hybrid and word-level. These features are extracted from signal and text (automatic transcriptions) representing the difficulty of the recognition. We experimented different features and learning algorithms in two scenarios: single hypothesis scenario and multiple hypotheses. For single channel (i.e. when there is only one automatic transcription for each utterance), we showed that the quality can be reasonably estimated through WER prediction using appropriate regression models based on extremely randomized tree (XRT). For multiple channel (i.e. when there are several transcriptions for each utterance, coming from several ASR systems or several microphones), we showed that machine-learned ranking (MLR) methods based on random forest (RF) provides a higher correlated ranking.

Chapter 4

Single Hypothesis ASR QE for Acoustic Model Adaptation

Chapter 3 introduced ASR QE in two scenarios based on the number of transcription channels: single hypothesis and multiple hypotheses. This chapter describes a novel application of single hypothesis ASR QE to improve unsupervised acoustic model adaptation. The method is based on applying pre-trained QE models on the output of the first decoding pass and use the QE results to inform the adaptation process about the quality of each adaptation sample. The next chapter is dedicated to the application of multiple hypotheses ASR QE.

4.1 Introduction

Acoustic model adaptation is necessary, in automatic speech recognition, because of the mismatches between the training and test sets. These mismatches are mainly due to speaker and accent variation, topic domain and like on. Acoustic model adaptation is a method to increase the robustness of the system towards these variations. Selecting the proper adaptation technique, however, depends on the size of the available adaptation data and the type of acoustic model.

This chapter explores different ways to enhance the adaptation method based on Kullback-Leiber divergense (KLD) (Yu et al., 2013) with automatic ASR quality predictions, described in §3.4. We focus on two alternative solutions for enhancing the adaptation:

1. *weighting* the KLD regularization term with coefficients that depend on the predicted quality of each transcribed sentence.
2. *filtering* the adaptation set by removing the utterances that, in terms of predicted quality, seem to be less reliable.

This is the first time that QE-based approaches are used for unsupervised DNN adaptation. The main contributions of this chapter are:

- a novel application of single hypothesis ASR QE to acoustic model adaptation;
- an extension to the KLD regularization approach for unsupervised DNN adaptation (Yu et al., 2013), which could be easily integrated in the KALDI speech recognition toolkit (Povey et al., 2011);
- significant improvement over hybrid DNN-HMM acoustic models.

After a short review on acoustic model adaptation methods in §4.2, a modified version of DNN-HMM acoustic model adaptation technique based on KLD regularization is presented in §4.3. The proposed method is evaluated through a range of experiments in §4.4. §4.4.3 includes the implementation details and §4.5 discusses the results. Finally in §4.7 we conclude this chapter.

4.2 Related work

Two popular adaptation techniques for GMM-HMM acoustic models (§2.3) are maximum likelihood linear regression (MLLR) and maximum a poste-

riori (MAP) (Wang et al., 2003b). MLLR computes a set of transformation matrices that will reduce the mismatch between an initial model and the adaptation data. In particular, MLLR estimates a set of linear transformations for the mean and variance parameters of the GMMs (Young et al., 2002). Each transformation matrix is applied on a specific class of parameters. MAP, instead, re-estimates the parameters of the model using the adaptation data. The sample mean/variance values are calculated over the adaptation data. and then the new mean/variance parameters of the model are updated toward the sample values. If the frequency of a phone is insufficient in the adaptation data, then the model of that phone does not change during adaptation (Wang et al., 2003b). Therefore, MAP requires larger adaptation data than MLLR.

Neither MAP nor MLLR are not completely suitable for DNN-HMM models, because these models usually contain a huge number of parameters to be adapted. A DNN-HMM acoustic model usually contains several hidden layers each including thousands of hidden neurons. This number of parameters makes the adaptation process a challenging task.

Several adaptation techniques have been proposed for artificial neural networks employed in ASR hybrid systems. These techniques are mostly based on the estimation of linear transformations for different layers including input, output or hidden layers (Gemello et al., 2007; Abrash et al., 1995; Neto et al., 1995; Li and Sim, 2010; Siniscalchi et al., 2013). Feature discriminative linear regression (fDLR) (Seide et al., 2011) and output-features discriminative linear regression (oDLR) (Yao et al., 2012) are two of these techniques. Regardless of the layer to which the transformation is applied, in these approaches, only the weights of the linear transformations are updated in order to optimize an objective function that is computed on the adaptation data. In this way, the risk that DNN overfits on the adaptation data is reduced.

A variant of fDLR is described in (Huang et al., 2014), proposing to adapt the DNN parameters within a maximum a posterior (MAP) framework. Basically, the method adds a term representing the prior density of the linear transformation weights to the objective function. This approach is demonstrated to be equivalent to L2 norm regularization (Li and Bilmes, 2006), if the prior distribution of transformation weights is Gaussian $\mathcal{N}(0, I)$. In general, adding a regularization term to the objective function has been proven to be effective for reducing the risk of overfitting.

Another adaptation technique that is applied to the input features is fMLLR (Parthasarathi et al., 2015). The difference between this method and fDLR relates to the criterion they adopt. fMLLR maximizes the likelihood of the adaptation data, while fDLR optimizes a discriminative criterion computed on the adaptation (e.g. it minimizes the mean squared error between target and actual output-state network distribution). Parthasarathi et al. (2015) observe that on clean speech data:

- filter-bank features and fMLLR features achieved comparable performance, and
- only the combination of the two types of features, either at an early or late fusion stage, provided significant WER reductions.

We investigate the impact of both filter-bank and fMLLR features on our proposed procedure for KLD adaptation.

In the context of speaker-adaptive training (SAT) via fMLLR (Gales, 1998), the recent approaches make use of i-vectors (Kenny et al., 2008) as speaker representation to perform acoustic feature normalization. Miao et al. (2015) train a neural network to convert i-vectors to speaker-dependent linear shifts and generate speaker-normalized features for training and decoding with SAT-DNN models. Garimella et al. (2015) proposes to process HMM-based i-vectors with specific hidden layers of DNN, before combin-

ing them with hidden layers. Karanasou et al. (2015) incorporates prior statistics (derived from gender clustering of training data) into i-vectors estimation. They show significant performance improvement when the approach is used for DNN adaptation in a hybrid ASR system.

In (Yu et al., 2013), Kullback-Leibler divergence (KLD) between the original distribution of the DNN outputs and the corresponding distribution estimated on the adaptation set is considered as regularization term. Yu et al. (2013) reports significant WER reduction compared to fDLR transformation. The weight assigned to the regularization term in the objective function is an important parameter that indicates when to use regularized learning. In this chapter, we maneuver on modifying this weight by means of predicted sentence-level WER scores.

Rather than the algorithm of the adaptation technique, the characteristics and quality of the adaptation data play a fundamental role. Pitz et al. (2000) shows that significant WER reduction is achievable by using confidence measures to remove the low confidence frames from the adaptation data. Thomas et al. (2013) propose an automatic sentence selection method based on different types of confidence measures for semi-supervised training of DNNs in a low-resource setting. However, the confidence measures are usually biased towards the decoder. From this point of view, they are not reliable enough for data selection. To address this issue, in this chapter we exploit single hypothesis ASR QE approach (§3.4) as a reliable data selection method to build the adaptation data.

4.3 KLD adaptation for DNN-HMM

KLD based adaptation can be implemented by adding a regularization component to the loss function in Equation 2.6. Yu et al. (2013) proposes to use the Kullback-Leibler divergence between the original distribution

and the adapted one as the regularization term.

Adding KLD regularization (that is computed on the adaptation data) to the cross-entropy formula results in the following objective function:

$$\mathcal{D}(\overset{*}{p}, p) = (1-\alpha)\mathcal{C}(\hat{p}, p) + \alpha\frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{I}\overset{*}{p}[s_i|o_t]\log p[s_i|o_t] \tag{4.1}$$

$C(\hat{p}, p)$ is the original objective function (Equation 2.6); $o_{t\in[1,N]}$ are the observation frames in the adaptation data; $\overset{*}{p}[s_i|o_t]$ is the posterior probability computed with the original DNN and α is the regularization coefficient. As reported in (Yu et al., 2013), Equation 4.1 can be rewritten as follows:

$$\mathcal{D}(\overset{*}{p}, p) = \frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{I}P[s_i|o_t]\log p[s_i|o_t] \tag{4.2}$$

where

$$P[s_i|o_t] = (1-\alpha)\hat{p}[s_i|o_t] + \alpha\overset{*}{p}[s_i|o_t] \quad 0 \leq \alpha \leq 1 \tag{4.3}$$

Equation 4.3 states that KLD regularization can be implemented through cross-entropy minimization between a new target probability distribution P and the current probability distribution p. The new target distribution is obtained as a linear interpolation of the original distribution $\overset{*}{p}$ and the distribution $\hat{p}$ computed via forced alignment with the adaptation data.

Note that, in Equation 4.3, $\alpha = 0$ is equivalent to do a "pure" retraining of the DNN over the adaptation data (i.e. completely trusting on the new data), while $\alpha = 1$ means that the output probability distribution of the adapted DNN is forced to follow the original DNN (i.e. completely trusting the original model). Usually, the value of α is estimated on a development set, together with the value of learning rate, and it does not change across the test utterances. One can expect that the optimal value of α is close

to 0 when the size of the adaptation set is large and the transcriptions of the adaptation sentences are not affected by errors (i.e. in supervised conditions). Otherwise, when the size of the adaptation set is small and/or its transcription can be affected by errors (i.e. in the case of unsupervised adaptation), the optimal value of α should increase.

Unlike other adaptation methods, KLD-based regularization binds directly the DNN output probabilities rather than the model parameters. In this way, the method can be easily implemented with any software tool based on back-propagation (e.g. the KALDI toolkit), with no modification.

4.3.1 Soft DNN adaptation

Yu et al. (2013) have shown a dependency of the optimal value of α on the size of the adaptation data. However, we believe that the optimal value of α depends not only on the size of adaptation data, but also on their quality. Starting from this intuition, we propose to compute α on a sentence basis, as a function of sentence-level WERs. Since in the real-life applications, the manual references are not available to compute the true WERs, we take advantage of automatic WER prediction using single hypothesis ASR QE method (§3.4).

In the proposed approach, the value of α is dynamically changed for each adaptation sentence. For example, for the k-th sentence, the value of α is defined as:

$$\alpha(k) = \text{pWER}_k, \qquad 1 \leq k \leq K \tag{4.4}$$

where, $0 \leq \text{pWER}_k \leq 1$ is an automatic prediction of the WER for the k-th sentence and K is the total number of adaptation sentences. Note that in this method, if the value of K is small and $\text{pWER}_k \cong 0, \forall k$, the original distribution $\overset{*}{p}$, in Equation 4.3, is weighted by $\alpha \cong 0$ (i.e. completely

trusting the adaptation data), augmenting the risk that the adapted DNN overfits the adaptation data. To avoid this effect we can simply add a bias to pWER as follows:

$$\alpha(k) = \beta + (1 - \beta) \times \text{pWER}_k \ , \quad 1 \leq k \leq K \ , \quad 0 \leq \beta \leq 1 \tag{4.5}$$

In the above equation, $\beta = 0$ yields $\alpha(k) = \text{predWER}_k$, i.e. the regularization coefficient depends only on the sentence transcription quality; $\beta = 1$ yields $\alpha(k) = \beta$, i.e. the regularization coefficient remains fixed over all adaptation sentences (this is the case of Equation 4.3). Therefore, optimizing over β allows us to control the trade-off between the quality of the supervision and the size of the adaptation set.

"soft" adaptation refers to DNN adaptation based on Equation 4.5, since the coefficients vary sentence by sentence, whereas "hard" adaptation refers to the method based on Equation 4.3, since the coefficients are fixed.

4.3.2 QE-informed data selection

The second usage of ASR QE is to select the adaptation data. This is mainly applicable to the unsupervised condition, when the manual reference of the adaptation data is not available. One example is to adapt the acoustic model to the test data, after the first decoding step. This is a common try to push the acoustic model towards the test domain. For this purpose, the test data is first recognized by the baseline acoustic model, then the QE procedure selects the hypotheses with high qualities (lower predicted WER) and finally the acoustic model is adapted to the selected data.

We use a similar method to the one described in §3.4 based on XRT regression models to predict the WER of the hypotheses after the first decoding step. Note that in this task, the acoustic models are known, so that

the glass-box features are available to train the QE models. Therefore, we use a combination of textual (black-box) and ASR decoder (glass-box) features to predict the sentence-level WER values. The black-box features are mainly the ones in Table 3.1. The glass-box features are mostly extracted from the confusion networks that are in turn obtained from the word lattices. Table 4.1 shows all the features that are used in the experiments.

ASR (9)	From each CN bin: the log of the first word posterior (1), the log of the first word posterior from the previous/next bin (2), the mean/std/min/max of the log posteriors in the bin (4), if the first word of the previous/next bin is silence (2)
Sentence level (10)	From each transcribed sentence: number of words (1), LM log probability (1), LM log probability of part of speech (POS) (1), log perplexity (1), LM log perplexity of POS (1), percentage (%) of numbers (1), % of tokens which do not contain only "[a-z]" (1), % of content words (1), % of nouns (1), % of verbs (1).
Word level (22)	From each transcribed word: Part-of-speech tag/score of the previous/current/next words (6), RNNLM probabilities given by models trained on in-domain/out-of-domain data (2), in-domain/out-of-domain 4-gram LM probability (2), number of phoneme classes including fricatives, liquids, nasals, stops and vowels (5), number of homophones (1), number of lexical neighbors (heteronyms) (1) binary features answering the three questions: "is the current word a stop word?"/"is the current word before/after repetition?"/"is the current word before/after silence?" (5).

Table 4.1: 41 features used for sentence-level WER prediction.

4.4 Experiments

4.4.1 Speech corpora

The speech data used in these experiments are collected for the 3rd CHiME challenge and it is publicly available.[1] For this dataset, six different microphones are placed on a tablet PC to record sentences of the Wall Street Journal (WSJ) corpus, uttered by noisy speakers in four different environments: bus, cafe, pedestrian area and street junction. We adopt three data from CHiME-3:

- *tr05_real* and *tr05_simu* used to train the baseline acoustic model;
- *dt05_real* used to train the QE model and
- *et05_real* used to test and compare the WER results.

tr05_real consists of 1,600 sentences uttered by 4 speakers in "real" noisy environment and *tr05_simu* consists of 7,138 sentence uttered by 83 speakers corrupted by "simulated" noise. *dt05_real (DT05)* contains 1,640 sentences uttered by four different speakers. *et05_real (ET05)* contains 1,320 sentences uttered by four other speakers. It is worth mentioning that, there is no speaker overlap between training, development and test sets. The number of utterances in the evaluation corpora is equally distributed among speakers and types of noise, that is, every speaker uttered the same number of sentences in each of the four noisy environments. In both training and evaluation data sets, utterance segmentation is done manually and the corresponding speaker identity is annotated. Therefore, no automatic speaker diarization module is employed in the experiments. Table 4.2 shows the statistics of these data[2].

[1] `http://spandh.dcs.shef.ac.uk/chime_challenge/download.html`.

[2] In addition, two parallel sets of "simulated" noisy utterances (namely *dt05_simu* and *et05_simu*) were generated as previously described. Though, we use only the real noisy data.

	tr05_simu	tr05_real	dt05_real	et05_real
duration	15h9m	2h54m	2h16m	1h50m
# sentences	7,138	1,600	1,640	1,320
# words	136.5k	28.3k	27.1k	21.4k
dict. size	8.9k	5.6k	1.6k	1.3k
# speakers	83	4	4	4
# noises	4	4	4	4

Table 4.2: Statistics of CHiME-3 training, development and test audio data.

4.4.2 ASR system

The architecture of the ASR systems are depicted in Figures 4.1 and 4.2. The former uses filter-bank features, while the latter uses fMLLR normalized features. The systems are mainly based on KALDI CHiME-3 v2 package, described in (Hori et al., 2015), with the addition of a second decoding pass that performs unsupervised DNN adaptation as described in §4.3.

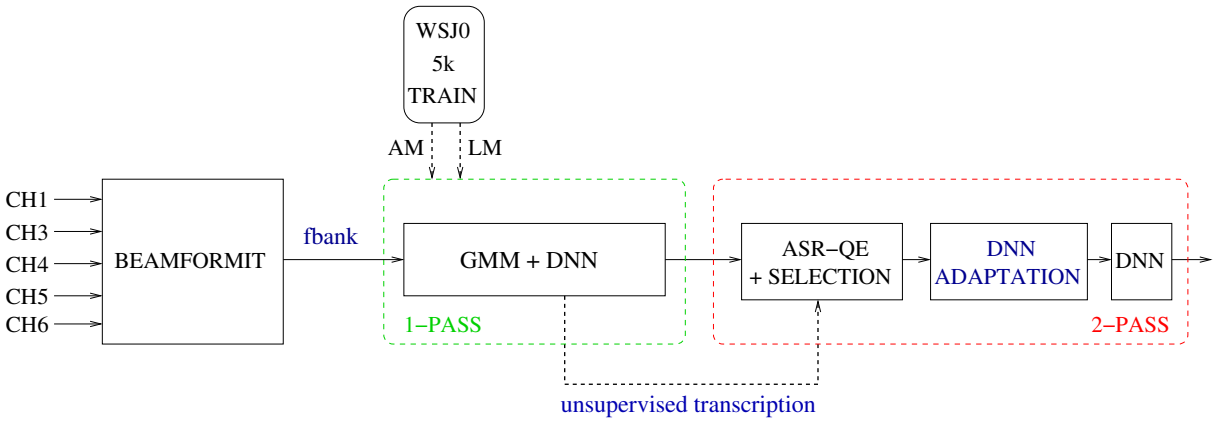


Figure 4.1: ASR architecture based on the KALDI CHiME-3 package with standard filter-bank features, plus QE hypotheses sorting and DNN adaptation.

A simple delay-and-sum (DS) beamforming consisting in uniform weighting of the rephased signals of the 5 frontal microphones is performed to enhance the quality of the signals. For this purpose the well known Beam-

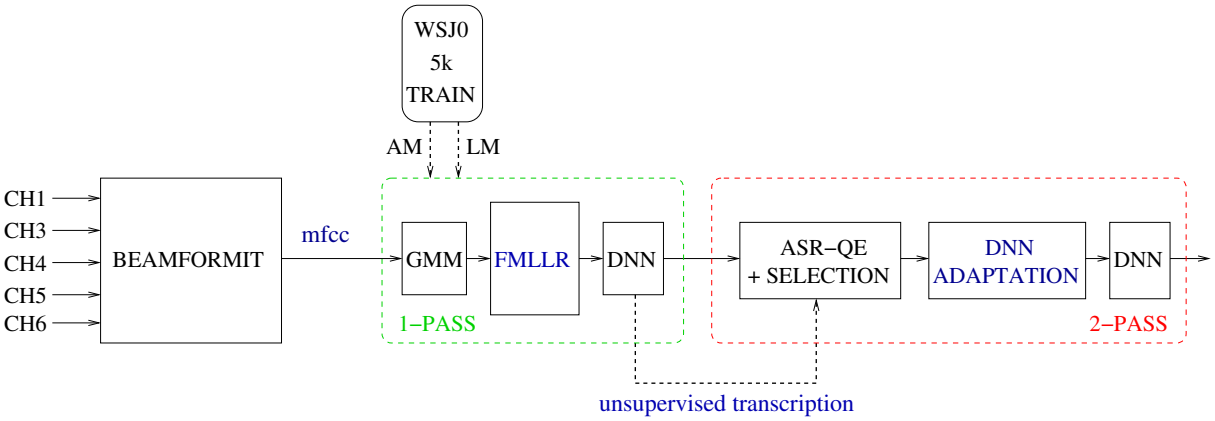


Figure 4.2: ASR architecture based on the KALDI CHiME-3 v2 package with fMLLR feature plus ASR QE hypotheses selection and DNN adaptation.

formIt toolkit (Anguera et al., 2007) is exploited. After beamforming, both filter-bank and fMLLR features are computed and processed by a corresponding hybrid DNN-HMM system that produces the supervision for adapting the DNN in the final decoding pass.

Filter-bank features

The employed filter-bank consists of 40 log Mel scaled filters. Feature vectors are computed every $10ms$ by using a Hamming window of $25ms$ length and they are mean/variance normalized on a speaker-by-speaker basis. The baseline DNN-HMM is trained using the Karel's setup (Vesely et al., 2011) in KALDI. To this aim the 8,738 training utterances are aligned to their transcriptions by means of the baseline GMM-HMM models.[3] An 11-frame context window (5 frames on each side) is used as input to form a 440 dimensional feature vector. The DNN has 7 hidden layers, each with 2,048 neurons. The DNN is trained in several stages including restricted Boltzmann machines (RBM) pre-training, mini-batch stochastic gradient descent training, and sequence-discriminative training using

[3]The initial GMM system makes use of the KALDI recipe associated to the earlier CHiME challenges (Barker et al., 2013; Vincent et al., 2013).

state-level Minimum Bayes Risk (sMBR). Initially, the learning rate is set to 0.008 and it is halved every time the relative difference in frame accuracy between two epochs on a cross-validation set falls below 0.5%. A frame accuracy improvement on the cross-validation set lower than 0.1% stops the optimization. All experiments involving adaptation of the baseline DNN, aiming at minimizing the objective function defined in Equation 4.2, are performed according to the above recipe.

fMLLR features

For this system, 13 mel-frequency cepstral coefficients (MFCCs) are computed every $10ms$ by using a Hamming window of $25ms$ length. These features are mean/variance normalized on a speaker-by-speaker basis, spliced by +/- 3 frames next to the central frame and projected down to 40 dimensions using linear discriminant analysis (LDA). Then, maximum likelihood linear transformation (MLLT) is applied and a single speaker-dependent fMLLR transform is estimated and applied to train speaker-adaptively trained (SAT) triphone HMMs.

During the first decoding pass, according to the given KALDI recipe, the computation of fMLLR features is done in two steps. First, a word graph is produced for each input utterance by using the baseline speaker-independent GMM-HMM. Then, a single fMLLR transform for each speaker is estimated from sufficient statistics collected from word graph with respect to SAT triphone HMMs. These transforms are used with SAT triphone HMMs to produce new word lattices. A second set of fMLLR transforms is estimated from new word lattices and combined with the first set of transforms. Finally, the resulting transforms are used to normalize the features processed by the hybrid system in the first decoding pass of Figure 4.2. The training of the corresponding baseline DNN, as well as DNN adaptation by KLD regularization, use the recipe adopted for filter-bank

features.

Language models

A 3-gram language model with Kneser-Ney smoothing method for estimating back-off probabilities is used for decoding. This model is trained with ≈37M words provided by the CHiME-3 organizers. After pruning low frequency words, the vocabulary size is ≈5K words. The perplexity value is 119.2, that is measured over *DT05* reference transcriptions.

Although not depicted in the figure, the n-best lists generated in the second decoding pass are rescored with a 5-gram LM and an RNNLM included in the CHiME-3 v2 package. Both models are again trained on ≈37M words provided by the CHiME-3 organizers.

4.4.3 Experimental setup

The soft adaptation approach described in §4.3 is applied in both "oracle" and "predicted" conditions. Oracle WER scores (*oWER* henceforth) are computed from reference transcriptions, while predicted WER scores (*pWER*) are estimated by the ASR QE system described in §4.3.2. Both values are used as WER estimates in Equation 4.5 to compute the target probability distribution. The performance achieved by oracle sentence WER represents the upper bound of the soft adaptation approach.

The regression models are trained on *DT05* and they are tuned with 8-fold cross validation, to minimize the MAE (Equation. 3.6) between the predicted and true WERs. 8-fold partitioning is done intentionally to avoid speaker and sentence overlaps between training and test folds.

Table 4.3 gives the list of DNN adaptation experiments. Each experiment is identified by: *i)* a combination of adaptation/evaluation sets; *ii)* the supervision used (manual or automatic); and *iii)* the features employed

(filter-bank or fMLLR normalized). For instance, the experiment named *DT05+man+fMLLR+ET05* in the first row indicates that the baseline DNN is adapted using *DT05* as adaptation set with the manual supervision and fMLLR features and it is evaluated on *ET05*.

type of experiment	adaptation set	type of supervision	features type	evaluation set
DT05+man+fMLLR+ET05	DT05	manual	fMLLR	ET05
DT05+man+fbank+ET05	DT05	manual	filter-bank	ET05
DT05+auto+fMLLR+DT05	DT05	automatic	fMLLR	DT05
DT05+auto+fbank+DT05	DT05	automatic	filter-bank	DT05
ET05+auto+fMLLR+ET05	ET05	automatic	fMLLR	ET05
ET05+auto+fbank+ET05	ET05	automatic	filter-bank	ET05

Table 4.3: List of DNN adaptation experiments.

Note that DNN adaptation with manual supervision (the first two rows of Table 4.3) is only meaningful in cross conditions, i.e. if adaptation and evaluation sets are distinct. The automatic supervisions of the adaptation sets (i.e *DT05* or *ET05*, depending on the experiment type) are produced by the first decoding pass as depicted in Figures 4.2 and 4.1. KLD regularization with manual supervision is applied according to Equation 4.3.

4.5 Results

The first set of results are achieved in *cross conditions*, meaning that the adaptation set is distinct from the evaluation set. The second set are obtained in *homogeneous conditions*, i.e. when the adaptation set coincides with the evaluation set. Finally, a discussion on data selection for estimating fMLLR transformations will conclude the section.

4.5.1 DNN adaptation in cross conditions

In this condition, we first use *all* the sentences in *DT05* for adapting the DNN. Then, we use a *subset* of the sentences in *DT05* which has been selected automatically with regard to the lowest predicted WER scores.

Using *all* the adaptation utterances

Figure 4.3a shows the WER results on *ET05* by varying the regularization coefficient α in Equation 4.3, when the filter-bank features are used. The Figure shows the results using both manual and automatic supervision. The horizontal line in the Figure corresponds to the baseline performance. In a similar way, the performance achieved with fMLLR features is shown in Figure 4.3b.

As it can be seen, the use of manual supervision, or equivalently the *supervised adaptation*, improves the performance in comparison to the baseline, with both types of features. In both cases, there is an intermediate optimal value of α in the interval $[0, 1]$, indicating that we should not totally trust neither the original model nor the adaptation data.[4] With the best value, we gain about 1% WER reduction, indicating the efficacy of the interpolation procedure expressed by Equation 4.3.

The substantial increase of WER value in Figure 4.3a at $\alpha = 0$ (both for supervised and unsupervised adaptation) indicates that data overfitting has probably occurred. The same behavior is not observed with supervised adaption using fMLLR features (see Figure 4.3b, where at $\alpha = 0$ no significant performance degradation is observed). This result can be explained by considering that fMLLR transformations allow re-

[4] As explained in Section 4.3, a value of $\alpha = 0$ corresponds to completely ignoring the contribution of the original DNN output distribution in the construction of the cross-entropy function (i.e. completely trusting the adaptation data), while a value $\alpha = 1$ forces the DNN parameters to follow those of the original distribution (i.e. we completely trust the original model).

ducing the acoustic mismatch between adaptation (*DT05*) and evaluation (*ET05*) sets. Figure 4.3b confirms this observation, where the curve labelled *DT05+man+fbank+ET05* is shifted towards the right part of the graph more than the corresponding curve *DT05+man+fMLLR+ET05*. This means that the adaptation procedure trusts the fMLLR normalized features more than the filter-bank ones.

Referring to Figure 4.3b, data overfitting (at $\alpha = 0$) instead occurs with unsupervised adaptation, as if the errors in the supervision acted similarly to an acoustic mismatch between adaptation and evaluation sets. These outcomes motivate deeper investigation on the impact of reducing the number of erroneous automatic transcriptions on the adaptation process.

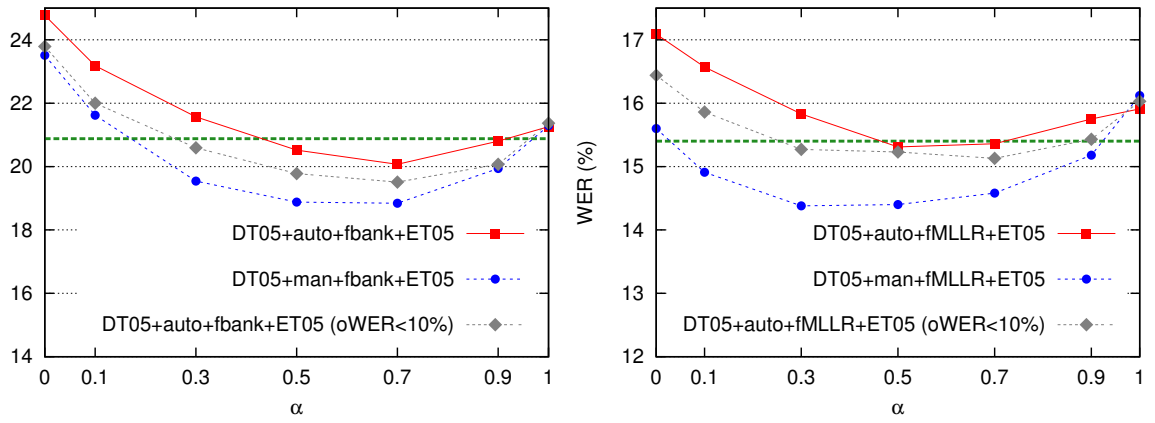
Selecting adaptation utterances

The idea is to remove the utterances whose WER is lower than 10% (the gray line, $DT05 + auto + * + ET05(oWER \leq 10\%)$) from the adaptation set (i.e. *DT05* in cross condition). Then, The baseline DNN is adapted with both hard and soft approaches (respectively Equation 4.3 and Equation 4.5). The results are shown in Figures 4.3. As it can be seen, the selection of adaptation utterances with WER$<$ 10% produces curves that approach those obtained using manual supervision, showing the benefits of reducing the transcription errors in the adaptation data.

In conclusion, DNN adaptation in cross condition can be performed "offline" on a development corpus (*DT05* in our case) and, hence, the resulting adapted DNN can be "immediately" used in an ASR system that employs only one pass of decoding.

4.5.2 DNN adaptation in homogeneous conditions

In this condition, the test set instances are used as the adaptation set for the second decoding pass. For this purpose, we conduct two decoding



(a) (b)

Figure 4.3: WER results achieved on evaluation set *ET05* as a function of the regularization coefficient α, using as adaptation set: the whole *DT05* (red and blue lines) and the subset of *DT05* with $oWER \leq 10\%$ (the gray line). The green lines indicates the baseline WER before adaptation.

passes,[5] as explained in §4.4.2. The output from the first decoding pass with the baseline DNN-HMMs provides the adaptation data for the successive adaptation steps. Then, the adapted DNN-HMMs are exploited in the second decoding pass to produce the final transcriptions.

Similar to the cross condition experiments, we first use *all* the sentences in *ET05* for adapting the DNN and then we use a *subset* of the sentences, selected automatically with regard to the predicted WER scores.

Using *all* the adaptation utterances

The results on both development and evaluation sets, with hard and soft adaptation strategies, are reported in Table 4.4. The numbers in parentheses show the absolute WER reduction with respect to baseline performance.

In the case of soft adaptation, we test both oracle and automatically-predicted WERs. Similar to the results in Figure 4.3b, the performance

[5]These experiments were motivated by the significant performance improvement obtained in (Jalalvand et al., 2015a) using "full" retraining of DNN in a two-pass ASR architecture.

experiment code	Baseline (no ada)	HARD ada	SOFT ada (*oWER*)	SOFT ada (*pWER*)
DT05+auto+fMLLR+DT05	8.2	8.0 (0.2)	7.9 (0.3)	8.0 (0.2)
DT05+auto+fbank+DT05	11.1	9.5 (1.6)	9.2 (1.9)	9.3 (1.8)
ET05+auto+fMLLR+ET05	15.4	14.5 (0.9)	14.3 (1.1)	14.4 (1.0)
ET05+auto+fbank+ET05	20.9	17.7 (3.2)	17.1 (3.8)	17.6 (3.3)

Table 4.4: WER results achieved by unsupervised DNN adaptation in homogeneous conditions. *ada*: α does not change; *QE-ada (oWER)*: α changes from one sentence to the other using the true WER values; *QE-ada (pWER)*: α changes using the predicted WERs

is measured as a function of the coefficients α and β. However, for clarity reasons, the whole set of results are not provided in Table 4.4. It mainly contains the top WER values achieved.

Unlike the results in Figures 4.3b, experiments in homogeneous conditions do not exhibit clear minimum values of the corresponding WER. Basically, no significant WER variations are observed for both α and β coefficients ranging in the interval $[0.0 - 0.7]$. The best performance is achieved for $\alpha = \beta = 0.7$, while for $(\alpha, \beta) > 0.7$ the WER increases.[6]

In Table 4.4, it is worth noting the significant WER reductions, compared to baseline, yielded by filter-bank features on both *DT05* and *ET05*. Although similar gains are not observed with fMLLR features, especially on *DT05* (as just pointed out above, probably due to their capability of reducing the acoustic mismatch between training and testing conditions), these results confirm the effectiveness of the two-pass decoding method. On the other hand, no substantial advantages are brought by the soft adaptation approach compared to the hard one. Despite this fact, it is worth observing

[6]To see examples of this trend of performance the reader can refer to the Figures 4.4 and 4.5 which report the WER scores achieved with hard adaptation and fMLLR features in homogeneous conditions (specifically, refer to the curves obtained without automatic sentence selection, respectively *DT05+auto+fMLLR+DT05* and *ET05+auto+fMLLR+ET05*).

the very close performance between oracle and predicted WER estimates, which demonstrates the efficacy of the proposed ASR QE approach.

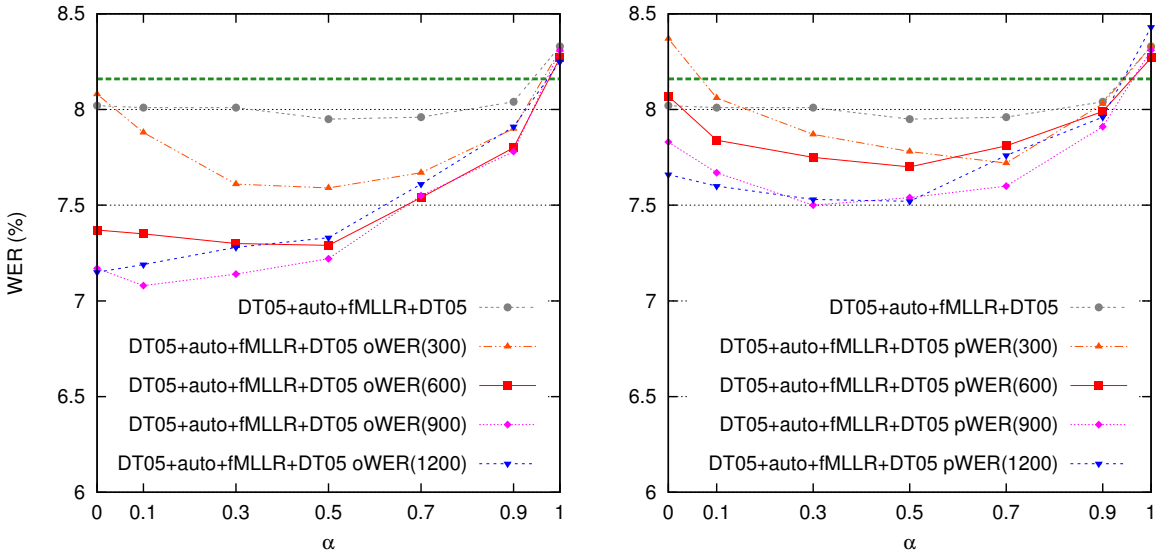
In summary, one can learn from these experiments that:

1. DNN adaptation in homogeneous conditions with two passes of decoding, using the whole set of adaptation utterances, yields performance improvements;

2. the selection of adaptation data based on oracle WER values is effective in cross condition and

3. no significant performance gain can be achieved with the soft adaptation method based on Equation 4.5.

The above observations, motivate further investigation aimed to find a criterion to automatically select the adaptation utterances in homogeneous conditions. A wise solution to choose such criterion is through ASR QE.



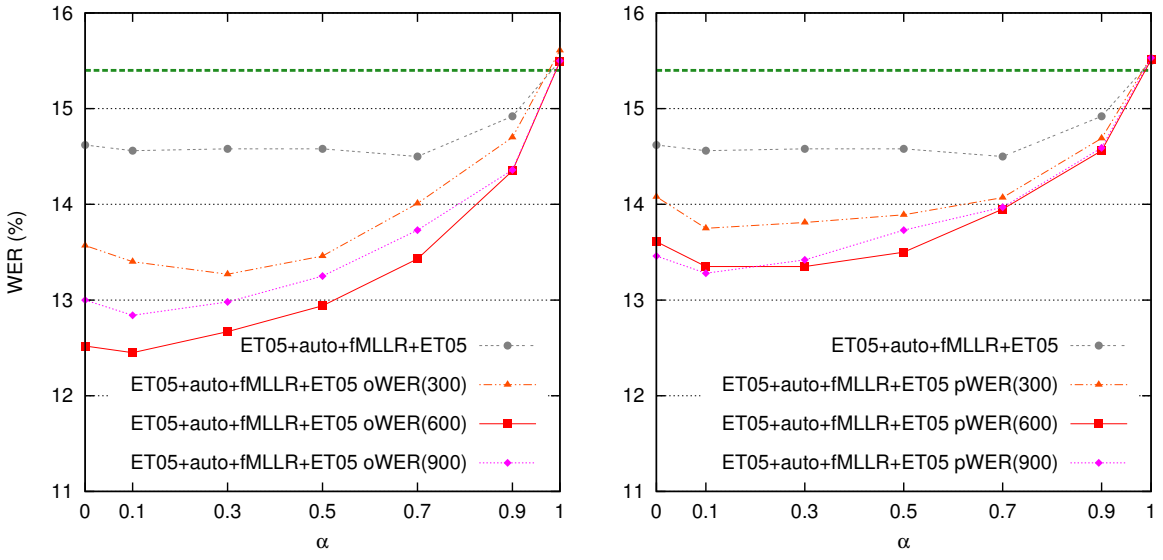
(a) (b)

Figure 4.4: WER results, achieved with oracle ($oWER$) and ASR QE ($pWER$) selection of adaptation utterances, on the development set *DT05*, as a function of the regularization coefficient α.

Using *automatically-selected* adaptation utterances

For the sake of clarity, the next set of experiments reports only the results obtained by fMLLR normalized features, since their WER is consistently lower than filter-bank features. However, the same and even more evident trends, were also observed using filter-bank features. Figure 4.4 reports the performance achieved on *DT05* using subsets of *DT05* with different sizes as adaptation data. The utterances of *DT05* are sorted according to the WER results from the first decoding pass. For sorting, we used both oracle WER values and the predicted ones obtained with the ASR QE (§4.3.2). From *DT05*, we extract four adaptation sets, respectively containing the "best" 300, 600, 900 and $1,200$ utterances. The various subsets, together with their automatic transcriptions, are used to adapt the baseline DNN by means of the hard adaptation approach. The reason for putting thresholds to the size of the adaptation set to compute the results lays in the fact that we want to make a fair comparison between the two selection methods (i.e by means of $oWER$ and $pWER$). In fact, sentence selection according to a preassigned WER threshold produces unbalanced adaptation sets of different sizes in correspondence to the application of each of the two methods.

From Figure 4.5, it is evident the efficacy of using only subsets of mid-high quality transcriptions for adapting the DNNs employed in the second decoding pass. Indeed, in each figure the minimum WER is reached with a couple of optimal values of the pair, (α, K), where K is the size of the adaptation set. This value is 900 for *DT05* (see Figure 4.4) and 600 for *ET05* (see Figure 4.5). The total improvement with respect to both the baseline performance and to the performance achieved using the whole set of adaptation utterances is remarkable. The difference in the optimal values of K for *DT05* and *ET05* is probably due to the different size of the two



(a) (b)

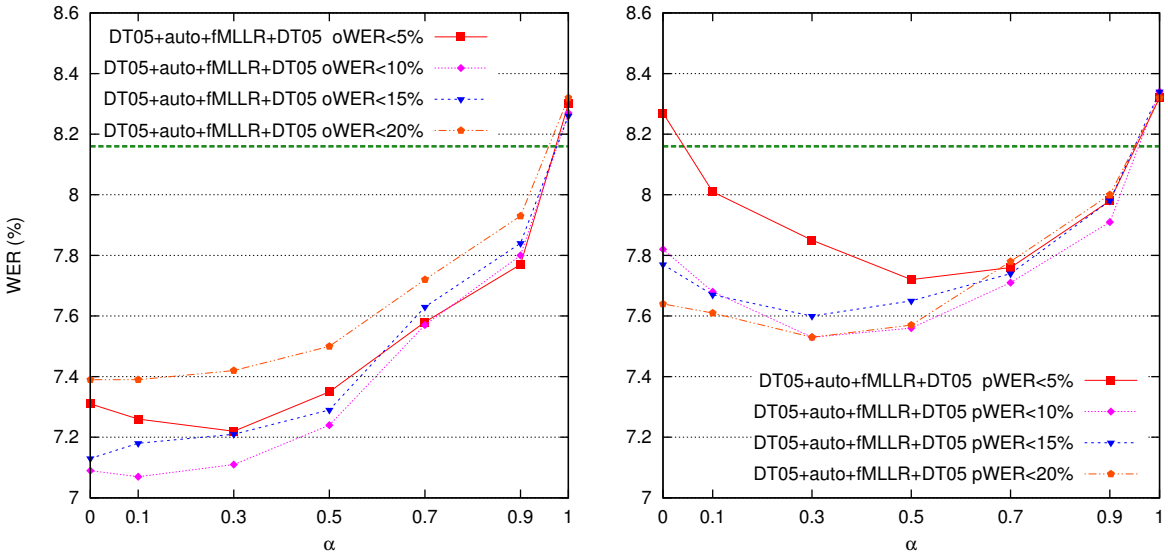
Figure 4.5: WER results, achieved with oracle (*oWER*) and ASR QE (*pWER*) selection of adaptation utterances, on the evaluation set *ET05*, as a function of the regularization coefficient α.

corpora (*DT05* contains 1,640 utterances, *ET05* contains 1,320 utterances). Unsurprisingly, the performance achieved with the ASR QE approach is lower than the upper-bound results obtained with oracle WER estimates. However, especially on the evaluation corpus *ET05*, the improvements over baseline performance are considerable.

In all figures, the optimal values for α are quite small, ranging in the interval $[0.1 - 0.3]$. This means that the adaptation method trusts more on adaptation set that now includes only "good" sentences.

Although for comparison purposes the analysis is focused on the size of the adaptation set to perform sentence selection, in real applications it is more feasible to select sentences on the basis of their predicted WER. Therefore, the next set of experiments was carried out by putting selection thresholds on WER predictions.

Figure 4.6a shows the performance achieved on *DT05* with hard DNN



(a) (b)

Figure 4.6: WER results, achieved with oracle ($oWER$) and ASR QE ($pWER$) selection of adaptation utterances, on the development set *DT05*, varying the WER thresholds.

adaptation as a function of α, varying the thresholds applied to $oWER$ values to select the adaptation utterances. Figure 4.6b, instead, shows the performance when the $pWER$ estimates are employed. Also in this case, the performance improvements with respect to the baseline, with both $oWER$ and $pWER$ are evident. The optimal values resulted to be $oWER_{thr} = 10\%$, $\alpha = 0.1$, where $oWER_{thr}$ indicates the selection threshold. Similarly, using $pWER$ estimates the corresponding optimal values are $pWER_{thr} = 10\%$, $\alpha = 0.3$. With $pWER$, the higher value for α compared to the value resulting from the use of the $oWER$ ($\alpha = 0.1$) approach is probably due to errors in the automatic WER predictions that have to be compensated.

Table 4.5 reports the final performance achieved on both *DT05* and *ET05* using the two-pass decoding approach and the automatic selection of adaptation utterances by means of automatically predicted WERs. For both sets, the optimal values of parameters α and $pWER_{thr}$ are those estimated on the *DT05* development corpus (i.e. $\alpha = 0.3$, $pWER_{thr} =$

10%).

	DT05		ET05	
	fMLLR	fbank	fMLLR	fbank
baseline	8.2	11.1	15.4	20.9
oWER	7.1	7.7	12.4	14.2
pWER	7.5	8.3	13.6	15.0

Table 4.5: WER results achieved using the optimal parameters (α and $pWER_{thr}$) estimated on *DT05*.

The results achieved by filter-bank features confirm the effectiveness of the proposed two-pass adaptation approach. Although filter-bank exhibits higher WER than the fMLLR baseline, after unsupervised adaptation the performance gap is significantly reduced (less than 2% absolute WER on *ET05*). In all cases, the small differences between the performance yielded by the use of oracle and the corresponding predicted WERs is remarkable.

The results in Table 4.5 are noticeable, considering that they outperform those given by a strong ASR baseline, implemented with state-of-the-art ASR technologies, i.e.: BeamformIt for speech enhancement, hybrid DNN-HMMs for acoustic modeling and speaker-dependent fMLLR transformations for acoustic model adaptation.

LM rescoring

Table 4.6 shows the results on *ET05* after LM rescoring procedure released in the updated CHiME-3 recipe. This procedure rescores the final word graphs produced in the second decoding pass by two consecutive steps: first by using a 5-grams LM, then by means of a linear combination of a 5-grams LM and a RNNLM.

The significant performance gains demonstrate the additive effect of LM rescoring over DNN adaptation and it allows us to reach a significant WER

	3-gram	5-gram	RNNLM
oWER	12.4	10.8	9.9
pWER	13.6	11.9	10.9

Table 4.6: WER results, achieved in homogeneous conditions on *ET05*, with automatic data selection and using the baseline LM rescoring passes (see Hori et al. (2015)).

of 10.9% on *ET05*.[7]

4.6 Discussion

The results in this chapter demonstrate that regardless of the type of acoustic features employed in the experiments (filter-bank or fMLLR normalized):

a) The benefits of KLD-based regularization are limited compared with DNN retraining without any regularization. This is probably due to the fact that the size of the adaptation sets considered in our experiments is large enough to prevent data overfitting (actually, previous research on KLD regularization Yu et al. (2013) demonstrates its effectiveness using only few minutes of adaptation data);

b) The presence of errors in the automatic transcription of the adaptation data is detrimental, especially when DNN adaptation is carried out in homogeneous conditions. In fact, comparing the results in the last two rows of Table 4.4 (achieved by using the whole *ET05* corpus as adaptation set) with those in Table 4.5 (obtained by using a subset of adaptation utterances with "few" transcription errors) we notice, in oracle conditions, absolute WER reductions of around 2% with

[7]See `http://spandh.dcs.shef.ac.uk/chime_challenge/results.html` for the official results of the challenge.

fMLLR and 4% with filter-bank features. Coherent WER reductions of around 1% and 3% are also achieved when applying QE-informed data selection. This demonstrates the effectiveness of the proposed QE-informed approach for DNN adaptation.

Till now, we have only considered KLD regularization for implementing DNN adaptation. However, as mentioned in Section 5.2, several previous works proposed alternative approaches based on the use of a single linear transformation, which can be applied either to the input or the output layer of the network. Therefore, in order to assess the effectiveness and the general applicability of the proposed QE-based approach, we also experimented with the output-feature discriminative linear regression (oDLR) transformation, in a way similar to that described in Yao et al. (2012). The results obtained in homogeneous conditions, both with and without ASR QE, are given in Table 4.7 (for comparison purposes, the baseline performance is also reported in the table). Similarly to results shown in Table 4.5, the optimal thresholds for both oracle and predicted sentence WER values are empirically estimated on *DT05*. The resulting values for $oWER_{thr}$ and $pWER_{thr}$ are respectively 10% and 20%.

	DT05		ET05	
	fMLLR	fbank	fMLLR	fbank
baseline	8.2	11.1	15.4	20.9
oDLR	7.9	9.6	13.8	17.5
oDLR+oWER	7.4	9.2	13.0	16.8
oDLR+pWER	7.7	9.5	13.6	17.2

Table 4.7: WER results, achieved in homogeneous conditions with oDLR-based adaptation, without using ASR QE (oDLR), using utterance selection based on oracle WERs (oDLR+oWER) and on predicted WERs (oDLR+pWER).

As shown in the table, the use of oDLR alone (even without ASR QE) always results in noticeable improvements over the baseline. The con-

siderable WER reductions measured in oracle conditions (oDLR+oWER), however, indicate the high potential of a QE-driven selection of the adaptation utterances also with this simple DNN adaptation method. In general, the performance improvements are smaller than the corresponding results for KLD regularization reported in Table 4.5. Such lower results can be explained by the findings reported in (Gollan and Bacchiani, 2008), in which the authors compared approaches based on MLLR and maximum a posterior probability (MAP) for GMM-HMMs adaptation. In this case, the impact of errors in the supervision is directly proportional to the number of transformation parameters to estimate. Indeed, while in the experiments reported in Section 4.5 all the parameters of the original DNN are adapted, with oDLR only a small fraction of them (around 13%) is updated. The reduced sensitivity to errors in the supervision is also reflected by the higher value of the threshold used to select the adaptation data (20% for oDLR vs 10% for KLD).

The results measured in oracle conditions suggest a higher potential for the application of QE to KLD-based regularization rather than to oDLR. This intuition, however, is partially contradicted by the last row of Table 4.7 (oDLR+pWER). With predicted WER scores, indeed, the values achieved with fMLLR are only slightly worse or identical to those in Table 4.5. To put into perspective this unexpected "exception" in the results, it's worth remarking that the impact of QE in DNN adaptation is proportional to the acoustic mismatch between training and test data. As observed in Sections 4.5.1 and 4.5.2, fMLLR features have the capability to reduce such mismatch, making the gains brought by QE-based adaptation less evident than those achieved with filter-bank. In light of this, although on *ET05* and with fMLLR features oDLR is competitive with the more complex KLD-based regularization proposed in this chapter, we believe that more challenging data (featuring a higher mismatch between

training and test) would increase the distance between the two approaches and reward our method.

4.7 Conclusions

This Chapter proposed to exploit single hypothesis ASR QE to perform unsupervised adaptation for deep neural network acoustic models (DNN-HMM). The main idea was motivated by the two following hypotheses:

1. The adaptation process does not necessarily require the supervision of a manually-transcribed development set. Manual supervision can be replaced by a two-pass decoding procedure, in which the evaluation data are automatically transcribed and used to inform the adaptation process;

2. The whole process can benefit from methods that take into account the quality of the supervision. In particular, automatic quality predictions can be used either to weigh the adaptation instances or to discard the less reliable ones.

To implement this approach, we retrained a (baseline) DNN by minimizing an objective function defined by a linear combination of the usual cross-entropy measure (evaluated on a given adaptation set) and a regularization term. This is the Kullback-Leibler divergence between the output distribution of the original DNN and the actual output distribution.

First, we experimented in "cross conditions", by adapting on the *real* development set of the CHiME-3 challenge and testing on the corresponding *real* evaluation set. In this scenario, we found that, when using all the manually-transcribed adaptation data, the KLD-based approach is effective. Then, moving to the automatically-generated supervision of the adaptation data, we discovered a correlation between performance results

and the quality of the adaptation data. In particular, in "oracle" conditions (i.e. with true WER scores), DNN adaptation benefits from removing utterances with a WER score above a given threshold.

Building on this result, we focused on "self" DNN adaptation in "homogeneous conditions", in which the baseline DNN is adapted on the same evaluation set (*ET05*) by exploiting the automatic supervision derived from a first ASR decoding pass. Similarly to the cross-condition scenario, this approach allowed us to significantly improve the performance when "low quality" sentences (i.e. sentences that exhibit oracle WERs higher than an optimal threshold) are removed from the adaptation set. Improvements were measured not only in "oracle" conditions (i.e. with true WER scores), but also in realistic conditions in which manual references are not available and the only viable solution is to rely upon predicted WERs. To this aim, we used automatic WER prediction as a criterion to isolate subsets of the adaptation data featuring variable quality. The results of an extensive set of experiments showed that:

- Exploiting ASR QE for DNN adaptation in a two-pass decoding architecture yields significant performance improvements over the strong, most recent CHiME-3 baseline;
- Self DNN adaptation is more effective with filter-bank acoustic features than with fMLLR normalized features. This behavior is probably due to the smaller mismatch between training and test data caused by the use of fMLLR transformations, indicating a higher potential of the QE-driven approach in a scenario characterized by weakness of fMLLR in reducing such mismatch (e.g. with small adaptation sets);
- ASR QE is less effective with output discriminative linear regression (oDLR) transformation for DNN adaptation, due to the lower number of parameters to adapt compared to KLD regularization. This

demonstrates the portability of our method, but a higher effectiveness with large DNNs.

Finally, we applied LM rescoring procedure to the word lattices produced after DNN-adapted decoding pass. The resulting WER reductions demonstrate the independent effects of LM rescoring and the proposed DNN adaptation approach. The last results showed that the proposed full-fledged system for DNN adaptation, integrating KLD and ASR QE for data selection, outperforms the strong CHiME-3 baseline with a 1.7% WER reduction (from 12.6% to 10.9%).

Chapter 5

Multiple Hypotheses ASR QE for System Combination

Chapter 3 introduced ASR QE in two scenarios based on the number of transcription channels: single hypothesis and multiple hypotheses. In Chapter 4, single hypothesis ASR QE was used to perform unsupervised DNN-HMM adaptation. This Chapter focuses on the application of multiple hypotheses ASR QE to improve the ASR output. The proposal of this chapter makes a significant contribution to ROVER, the most popular ASR system combination method, by using ASR QE for ordering the input components. The proposed method, named segment-level QE-informed ROVER yields salient WER reduction in two different tasks: combination of multiple ASR systems (IWSLT) and combination of multiple distant microphones (CHiME-3).

5.1 Introduction

In order to obtain more accurate transcriptions, ASR systems with sufficient diversity and complementarity are combined in different ways (Audhkhasi et al., 2014). The combination of multiple hypotheses coming from independent sources usually leads to significant improvement compared to

each individual system. ROVER (Recognizer Output Voting Error Reduction), a popular ASR system combination approach, performs hypotheses fusion by first building a word confusion network (CN) from the *1*-best hypotheses and then selecting the best word in each CN bin via majority voting (Fiscus, 1997). When available, word confidence scores are also used for weighted majority voting. This general strategy has been improved in several ways, but, despite their proven effectiveness, ROVER and its variants still have some potential drawbacks.

The first drawback of ROVER is intrinsic to its implementation. The fusion process starts from the first input hypothesis, which is used as a "skeleton" for the greedy alignment of the others. The order in which the hypotheses are used to feed the process can hence determine significant variations in the quality of the resulting combination. **This calls for automatic methods for ranking the hypotheses to initialise the fusion process.**

The second drawback is inherent to the way ROVER is normally run. The fusion process is typically fed with transcriptions of entire audio recordings (lasting up to several hours). With this level of granularity, the skeleton used as basis for the alignment may consist of long transcriptions whose quality can considerably vary at local level. For instance, the worst transcription of an entire audio recording (globally) could be the best one for some passages (locally). **This calls for solutions capable to operate at higher granularity levels (*e.g.* segments, lasting up to few seconds) to better exploit the local diversity of the input hypotheses.**

The third drawback relates to the applicability of ROVER-like fusion methods, because their common trait is the reliance on information about the inner workings of the combined systems. Indeed, the standard voting scheme weighted with confidence scores is usually much more reliable than

the simple frequency-based voting. **The access to confidence scores, however, is a too rigid constraint in application scenarios where the hypotheses to be combined come from unknown "black-box" systems**.

Finally, it is worth noting that confidence scores proposed by previous ASR literature (Evermann and Woodland, 2000; Wessel et al., 2001), even if applicable, only indicate how confident the system is about its own output (§2.5.2). This can be a biased perspective (influenced by individual decoder features), producing scores that are not comparable across different systems. **External and system-independent measures of goodness would represent a more reliable alternative when comparable and objective ASR quality judgements are required**.

To cope with these issues, in this chapter, we present *QE-informed ROVER* (segment-level quality estimation informed ROVER). In this approach, before starting the fusion process by ROVER, we rank the input hypotheses at segment-level (addressing the first and second issues) using ASR QE approaches described in §3.2 (addressing the third issue).

The performance of QE-informed ROVER is assessed on two distinct scenarios, confirming the generality of the proposed method. In the first scenario, we apply QE-informed ROVER to combine the automatic transcriptions of English TED talks generated by eight ASR systems in the IWSLT2013 evaluation campaign.[1] The proposed solution outperforms standard ROVER and it significantly approaches oracle upper-bounds.

The second scenario involves multiple distant microphone (MDM) speech recognition in noisy environments. The experiments are carried out with the data delivered for the 3^{rd} CHiME challenge,[2] also described in §4.4.1.

[1]The International Workshop on Spoken Language Translation (IWSLT – `http://workshop2013.iwslt.org/`) is a yearly workshop associated with an open evaluation campaign on spoken language translation.

[2]The CHiME Speech Separation and Recognition Challenges are international initiatives proposed in

Although MDM hypothesis combination can be considered as an alternative to the signal enhancement techniques (Wölfel and McDonough, 2009; Kumatani et al., 2011; Mestre and Lagunas, 2003), in this work we show that QE-informed ROVER yields further improvements even after including the enhanced channels in the combination.

To provide the QE results for ROVER, two strategies are utilized:

1. *ranking by regression* (§3.5.1). To predict the quality of each individual transcription channel and then rank them in order to be combined with ROVER.

2. *machine-learned ranking* (§3.5.2). To directly predict the ranks through pairwise comparison.

This is the first time that ASR QE is applied to rank MDM hypotheses.

Deeper analyses address the problem of tied ranks. This happens when there are several hypotheses with identical WER scores for a speech segment and consequently they obtain the same ranks, in spite of their different quality (McSherry and Najork, 2008). The existence of tied ranks in the training data can considerably degrade the performance of ranking machines. We show that breaking the ties according to the overall performance of each individual component (predicted by ASR QE), improves the ranking process.

The last experiments address the problem of finding the optimum level of combination. The optimum level of combination can be obtained by using simple classifiers trained on the features representing the diversity of the components.

The contributions of this chapter can be summarized as follows:

- test the impact of ROVER at higher granularity level (e.x. segments lasting up to a few seconds);

2011, 2013 and 2015 – `http://spandh.dcs.shef.ac.uk/chime_challenge/`.

- inform ROVER of the hypotheses order according to their predicted quality;
- tackle the problem of tied ranks in multiple hypotheses ASR QE;
- tackle the problem of finding the optimum level of combination.

5.2 Related work

The most relevant research strands to this work are ASR system combination and multiple distant microphone speech recognition.

System combination. Many approaches proposed in the past for combining multiple ASR outputs make use of word graphs (Li et al., 2002; Bougares et al., 2013). The underlying idea is to merge the word graphs generated by different ASR systems into a single one, which is then traversed to search for the best path. As an alternative, frame-based system combination (Hoffmeister et al., 2006) tries to minimize a cost function called "time frame word error" (fWER) over a set of word graphs produced by different ASR systems. The method makes it possible to estimate the path exhibiting the minimum Bayes risk, without the necessity of merging edges of single word graphs. Confusion network combination (CNC) (Mangu et al., 2000; Hoffmeister et al., 2006; Evermann and Woodland, 2000) is another widely investigated approach, in which confusion networks built from the individual lattices are aligned instead of single best outputs. Hoffmeister et al. (2007) makes a comparison between graph-based and 1-best combination approaches and they conclude that, for pairwise combinations, the graph-based approaches can outperform 1-best ROVER, but 1-best ROVER results are equal (or even better) when combining three or four systems.

Another system combination method is joint decoding that indeed combines the acoustic models. The acoustic models can be different in terms of training features, learning criteria and like on, but they share the same HMM topology. This method treats the acoustic models as separate streams and it makes a linear combination between the log-likelihoods computed by each model (Yang et al., 2016). Since this approach requires to know the inner information of the acoustic models, it does not fit to our condition in which the ASR systems are not known.

Note that all combination methods based on the use of word graphs, as well as some extensions of ROVER (Schwenk and Gauvain, 2000; Zhang and Rudnicky, 2006; Hillard et al., 2007; Abida et al., 2011), require to know and have access to the inner structure of ASR decoder. Whereas, usually ASR systems, especially those embedded in commercial applications, do not provide this information. Instead, standard ROVER and QE-informed ROVER do not necessarily need to know the ASR system characteristics.

Another limitation of previous ROVER-based methods is that, assuming a fixed order in the quality of ASR systems, they do not apply any segment-level ranking. This leads to disregard the possible advantages of operating at a higher granularity level. Instead, in QE-informed ROVER, we increase the level of granularity to the length of the segments that are usually defined automatically according to some criteria such as pauses, speaker switches and like on.

Multiple Distant Microphone (MDM) speech recognition. Using multiple microphones for recording the speech signal from different angles, helps in tackling the fundamental problems such as noise, echo and reverbration. The MDM task usually involves several modules such as noise cancellation (Benesty et al., 2009), channel selection (Kumatani et al., 2011; Wölfel and

McDonough, 2009), signal-level combination (Mestre and Lagunas, 2003) and hypothesis-level combination (Barker et al., 2015; Guerrero and Omologo, 2014). Noise and echo cancellation, as well as channel selection, have been widely studied in signal processing literature. Concerning channel selection, one of the simplest approaches is to compute the signal-to-noise ratio (SNR). The channel with the highest SNR would be easier to transcribe (Wölfel and McDonough, 2009). Another solution is to consider the confidence measure (§2.5.2) provided by the ASR decoder (Jalalvand et al., 2015a). Signal-level combination, or beamforming, refers to a set of methods to generate an enhanced signal from multiple recordings. To this aim, minimum variance distortionless response (MVDR) and delay-and-sum (DS) are well known algorithms (Wölfel and McDonough, 2009; Barker et al., 2015; Kumatani et al., 2012) to enhance the signals.

For the hypothesis-level combination in MDM task, Wölfel and McDonough (2005) propose the use of CNC and Stolcke (2011) extends it with a hybrid approach that leverages beamforming and signal-level diversity. Guerrero and Omologo (2014) use inter-microphone agreement to build a confusion network from multiple word graphs to improve ASR in a domotic application.

This Chapter applies ASR QE and machine-learned ranking in the MDM scenario to perform microphone combination at hypothesis level. The results of the experiments confirm that hypothesis-level combination yields further WER reduction on top of signal-level combination, thanks to the proposed QE-informed ROVER.

5.3 ROVER

Figure 5.1 shows the architecture of ROVER with two main modules: alignment and voting. The alignment module combines the 1-best hypotheses

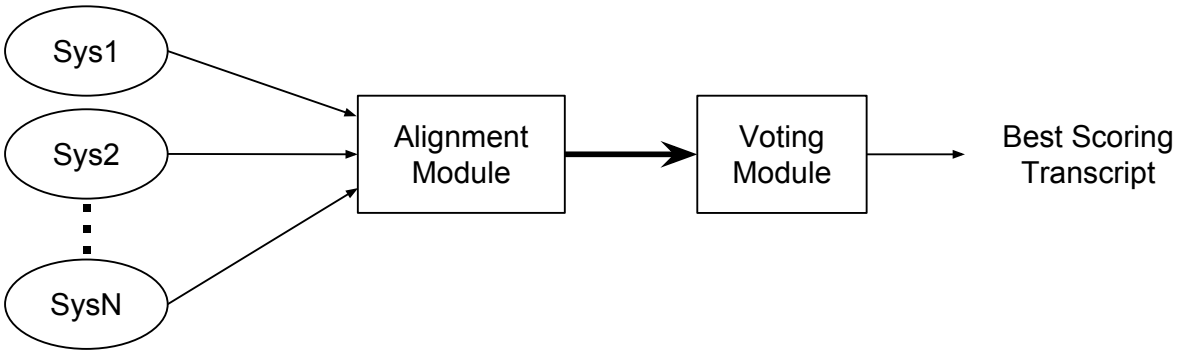


Figure 5.1: ROVER system architecture

from two or more ASR systems into a word transition network. The network is created using iterative dynamic programming alignment. This network can be shown as a confusion network consisting of a series of slots. Each slot is consisting of several arcs. Each arc corresponds to a word (and a confidence score, if available) recognized by one of the ASR systems. Voting module evaluates the individual slots using a voting scheme. This module selects the best word according to a posterior score that is computed by weighted interpolation between vote numbers and confidences:

$$score(w,i) = \sum_{l=1}^{L} \lambda_l [\alpha.\delta(w, w_l^i) + (1-\alpha).conf_l(w,i)]$$

$score(w,i)$ is the score of the word w at the i-th slot of the network; δ is the Kronecker-δ; L is the number of systems; w_l^i is the word suggested by l-th system at slot i and $conf_l$ indicates the confidence score of the l-th system. System votes and confidence scores are smoothly interpolated via α. The importance of the systems can be weighted by λ. However, in basic ROVER $\lambda_1 = ... = \lambda_L = 1/L$ (Hoffmeister et al., 2007). Note that in this chapter, the ASR decoders' information including confidence scores are not known. Therefore in the above formula, we always consider $\alpha = 1$.

Figure 5.2 shows an example of basic ROVER when there are three

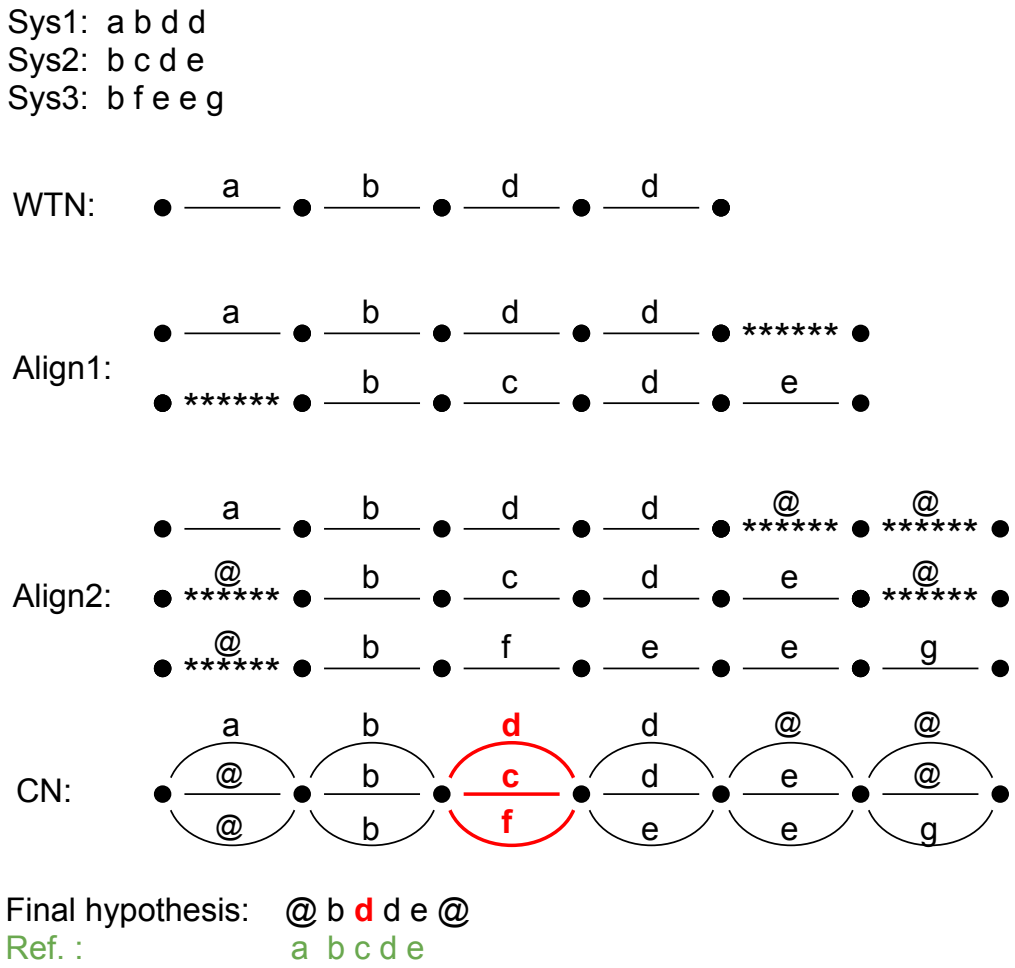


Figure 5.2: ROVER procedure. *Sys*2 has recognized better than *Sys*1 though it is in the second order. This mistake in the input arrangement, leads to an error in the final hypothesis, while it could be recovered with correct arrangement.

hypotheses to be combined. As it can be seen, ROVER starts by creating a WTN on the first entering hypothesis ($WTN1$) as the skeleton of the combination. The second and third solutions are aligned via a Levenshtein alignment with the time overlap between the words as a local cost. The final WTN can be shown as a confusion network (CN) (Mangu et al., 2000) with a sequence of slots. For each slot the best word is selected via majority voting, yielding the final hypothesis.

In case different words receive the same vote (as the red slot in Figure 5.2), ROVER gives priority to the word suggested by the first entering input. This is exactly where ASR QE can play an important role by ordering the inputs, descending the quality.

5.4 QE-informed ROVER

Traditionally, ROVER is applied using several random orders or blind fixed orders at system-level (Cettolo et al., 2013). By *system-level* ranking, we refer to an ordering that is assigned to component systems. This ranking is the same for all the segments in the dataset. Instead, *segment-level* ranking dynamically varies from one segment to another, aiming to take advantage from the fact that different ASR systems (or microphones in MDM scenario) show different accuracy for each segment.

QE-informed ROVER, differently from the traditional ROVER and its extensions like iROVER (Hillard et al., 2007) and cROVER (Abida et al., 2011), combines the hypotheses at segment-level and it precisely orders the inputs before performing alignment. This is done by exploiting multiple hypotheses ASR QE approaches (§3.5).

To investigate the effectiveness of segment-based ROVER, we run some pilot experiments on two distinct datasets, namely: *IWSLT*, consisting of English TED talks transcriptions (collected with a close talk microphone)

and *CHiME3*, previously described in §4.4.1. In the first case, we use the submissions to the IWSLT2013 ASR evaluation campaign (Cettolo et al., 2013), in which 8 teams submitted their ASR results. In the second case, we use the data provided for the 3^{rd} CHiME challenge (Barker et al., 2015), where each utterance is recorded by five microphones embedded on a tablet PC. CHiME-3 recordings are transcribed using two baseline ASR systems provided by the organizers: one employing Gaussian mixtures for computing acoustic probabilities, and the other using deep neural networks. Therefore, in the first task (*IWSLT*), we combine up to 8 different hypotheses, while in the second task (*CHiME-3*), we combine up to 10 different hypotheses.

With the following analysis, the first goal is to check if, and to what extent, an oracle ranking at system-/segment-level can positively contribute to ROVER results. The results are achieved by running the basic ROVER with different input orders.

Ranking	IWSLT (best WER=13.5%)				CHiME-3 (best WER=32.6%)			
	L1	L3	L5	L8	L1	L3	L5	L10
SysO	13.5	12.2	**11.8**	12.1	32.6	30.7	**27.9**	29.4
SegO	**8.9**	10.5	11.4	11.7	**19.5**	21.4	22.4	28.4
InSysO	27.2	19.8	15.1	13.3	40.5	37.5	34.0	29.7
InSegO	33.8	22.9	17.4	13.0	56.3	51.0	42.6	30.9

Table 5.1: WER results of ranking methods on IWSLT and CHiME-3 test data. In IWSLT, the best individual system results in 13.5% WER. In CHiME3, the best system (5th microphone transcribed by DNN model) results in 32.6% WER.

In Table 5.1, *SysO* and *SegO* represent the WER scores achievable respectively at system- and segment-level, thanks to ideal oracle ranking (derived from the exact WER of each ASR hypothesis with respect to manual references). The WER values are shown at different levels of combination in columns: $L1$ means that there is only one system (no combination); $L3$

means that there are three systems combined together and so on. At both granularity levels, oracle rankings are derived from the true WER scores of the candidate transcriptions. As it can be seen from the table, oracle-based hypotheses combination allows for considerable WER reductions compared to the best individual system, at system- and segment-level. In particular, on IWSLT, system-level combination results in a WER reduction from 13.5% (best individual system) to 11.8% when combining the best 5 systems. Similarly, on CHiME-3, WER decreases from 32.6% (best individual channel[3]) to 27.9% when the best 5 hypotheses are taken. Note that, in both tasks, system-level combination achieves the best results with only 5 different hypotheses, which is less than all the available ones.[4] This can be explained by observing that all systems/channels contribute to the final voting with the same weight. Therefore, the insertion of the worst hypotheses in the ranked list contributes to worsen rather than to improve the global performance.

Segment-level combination gives significantly better results than system-level combination in both tasks. In SegO, the WER increases by augmenting the number of hypotheses, because of entering erroneous hypotheses in the combination. Actually, being able to correctly select the best hypothesis for each segment (SegO - *L1*) forms the highest performance on both tasks, respectively 8.9% on IWSLT and 19.5% on CHiME-3.

The last two rows of Table 5.1 show the performance of ROVER when the transcriptions are combined in inverse oracle order, from the worst to the best one, at both system- (InSysO) and segment-level (InSegO). The poor results achieved, especially at lower combination levels, demonstrate that the rank of the input hypotheses is critical in ROVER combination

[3]Henceforth, for CHiME-3 data, the terms "channel" and "microphone" will be interchangeably used.

[4]The fact that on both datasets the best level of combination is the same is purely incidental. As we will see in §5.6.2, the definition of the optimum level of combination for a given dataset is an interesting research direction, which is partially explored also in this work.

results.

By this analysis, the impact of QE-informed segment-level ROVER is visible. Now it is interesting to investigate how much and to what extent *multiple hypotheses ASR QE* (§3.5) can approximate the oracle rankings.

5.5 Experiments

In the experiments, we use both strategies:

- *ranking by regression (RR)* §3.5.1. This strategy adopts single hypothesis ASR QE for multiple hypotheses purpose. That is, the WER of all the hypotheses are predicted individually and then they are ranked according to the predicted values. Afterwards, ROVER combines the hypotheses.

- *machine-learned ranking (MLR)* §3.5.2. This strategy adopts ranking algorithms (Cao et al., 2007) to directly predict the ranks using pairwise comparison.

It's important to remark that the results reported in this work are not comparable with the official submissions to IWSLT and CHiME-3 challenges (Barker et al., 2015; Jalalvand et al., 2015a), since here we assume that neither the ASR systems nor their confidence scores are accessible, whereas the official submissions take advantage of a range of techniques such as acoustic model adaptation, confidence-based data selection and multi-pass recognition as well as language model rescoring. In the following experiments, we do not consider these techniques and the QE approaches are applied to the first pass decoding output of the baseline systems (Barker et al., 2015). This is to confirm that the proposed method is general enough to be applied in a variety of black-box conditions.

5.5.1 Features

All the features described in Table 3.1 are used for this task. These features are grouped into three sets:

- **Basic (B)**: the combination of Signal, Textual and Hybrid features from Table 3.1.
- **Word-based (W)**: word-level features described in Table 3.1.
- **Basic+Word (BW)**: the combination of all features.

The reason for this grouping is to compare the baseline features previously introduce in Negri et al. (2014) with the newly proposed word-based features that are inspired by the works related to word error detection (Goldwater et al., 2010; Tam et al., 2014; Jalalvand and Falavigna, 2015).

5.5.2 Terms of comparison

The results are compared with the standard random ROVER and the two oracle systems described in §5.4.

- **Random ROVER.** The entire transcription candidates obtained from different systems/channels are taken in random order.
- **System-level oracle (SysO).** The true overall rank for each ASR system (or microphone) is known and this rank is kept unchanged for all the segments.
- **Segment-level oracle (SegO).** The exact rank of the ASR hypotheses at segment-level is known and ROVER is applied segment by segment using the true ranks. The result obtained in this way is the strongest term of comparison.

The last two terms are considered as "oracles" because they rely on external information about the true WERs, which is not accessible in real-life applications. The first goal is to significantly outperform random ROVER, however reducing the performance gap with respect to the two oracles would represent an even more convincing measure of success.

5.5.3 IWSLT task

Data. For this task, we use the submissions to IWSLT2012 and IWSLT2013 ASR evaluation campaigns respectively as training and test sets. Both sets are collected from English TED talks dealing with different topics. Six groups participated in IWSLT2012, in which the best performance (12.4% WER) was obtained by NICT group[5]. In IWSLT2013 two other groups participated. Also in that year, NICT won the challenge by 13.5% WER. Tables 5.2 and 5.3 provide some basic statistics about this dataset and the individual WER results. Complete details about each ASR system can be found in (Federico et al., 2012) for 2012 and (Cettolo et al., 2013) for 2013.

By selecting the data from two different years, we guarantee that there is no speaker nor topic overlap between training and test sets. Although most of the submissions of 2013 come from the same laboratories of 2012 (except for the two teams that did not participate in 2012), the ASR systems are quite different due to the changes and the improvements made by participants during the course of one year of research. Finally, in this audio dataset, since the recordings are carried out by head-mounted microphones, there is not any reverberation or background noise apart from the applause breaks, especially in the final part of the talks. However, it happens frequently that the speakers, sometimes non-native ones, make spontaneous speech phenomena such as hesitations, repetitions and false starts during their talks.

[5] https://www.nict.go.jp/en/

Attributes	IWSLT2012 (training)	IWSLT2013 (test)
duration (hr)	1h45m	4h50m
# sent	1,124	2,246
# token	19.2k	41.6k
dict. size	2.8k	5.6k
# speakers	11	28

Table 5.2: Statistics of IWSLT task

System	IWSLT2012	IWSLT2013
FBK	16.8	23.2
KIT	12.7	14.4
MITLL	13.3	15.9
NAIST	–	16.2
NICT	12.4	13.5
PRKE	–	27.2
RWTH	13.6	16.0
UEDIN	14.4	22.1
Avg.	13.86	18.56

Table 5.3: WER results of individual ASR systems in IWSLT task

In short, the experiments on this task are conducted as follows:

1. Utterance segmentation of IWSLT2012 (training) and IWSLT2013 (test) data;
2. Feature extraction from the $(signal, transcription)$ training pairs;
3. Training of the QE models (regressors and ranking machines);
4. Feature extraction from the $(signal, transcription)$ test pairs;
5. Estimating the WERs/ranks of the test hypotheses; and concatenation of the resulting outputs;
6. Computation of the WER scores.

While for IWSLT2012, manual utterance segmentation is provided and shared among the participants, for IWSLT2013, the segmentation had to be carried out automatically by each individual ASR system before decoding the audio tracks. Since each system produces a different number of segments, it becomes necessary to align each automatic segmentation with a reference one in order to share the same utterance time boundaries

Ranking	L3	L4	L5	L6	Avg. Impr.
Random	13.4	11.8	12.3	11.8	0.0
SysO	11.4	9.3	9.6	9.5	-2.4
SegO	8.0	7.9	8.2	9.1	-4.0
RR1+B	11.5	10.2	9.9	9.8	-2.0
RR1+W	13.2	10.6	10.1	9.8	-1.4
RR1+BW	12.1	10.3	10.0	9.8	-1.8
RR2+B	11.3	10.3	9.9	9.8	-2.0
RR2+W	12.1	10.3	10.0	9.8	-1.8
RR2+BW	11.2	9.9	9.8	**9.6**	-2.2
MLR+B	10.7	9.8	9.7	**9.6**	-2.4
MLR+W	10.7	**9.7**	9.7	**9.6**	-2.4
MLR+BW	**10.6**	9.8	**9.6**	**9.6**	-2.4

Table 5.4: WER results of different ranking methods on IWSLT2012 using 4-fold CV. *L6* indicates that the output of all the 6 systems participated in IWSLT2012 are combined

among the different hypotheses. In principle the segmentation given by one randomly chosen ASR system is taken as reference.

Preliminary analysis. To test the performance of our method in a controlled setting, we first run QE-informed ROVER on the training set (IWSLT2012) using 4-fold cross validation. Data partitioning is done considering the speakers in order to guarantee that there is no speaker overlap between training and test folds. Moreover, the partitioning is done in such a way that each instance occurs only once in each test fold. Thus, by collecting the results from all folds, we obtain the performance on the whole training data. Therefore, the numbers reported in Table 5.4 refer to the WER of the whole IWSLT2012.

The first three rows of the table show the results of ROVER when the input hypotheses are randomly ordered at system level and those achieved by the two oracles (SysO and SegO). Even if Random ROVER achieves

better results than the individual IWSLT2012 participants (see Table 5.3), it is quite far from the oracles. SysO is significantly better, with an average WER reduction of 2.4% over Random ROVER. As expected, the best performance is achieved by SegO with an average WER reduction of 4.0%. These differences confirm that running ROVER without an optimal order of the hypotheses may limit the gain achievable by system combination. Apart from that, both oracles reach the lowest WER score at combination level $L4$. This indicates that the majority of the aligned words in the confusion sets of the word network built by ROVER falls in the top four hypotheses. Inserting more alternative words from less accurate hypotheses at $L5$ and $L6$ has a negative impact on majority voting.

Ranking by regression, using all the training transcriptions and the basic features (RR1+B) already outperforms Random ROVER but remains far from the oracles. Slightly worse performance is obtained when using the word-based features (RR1+W) and the union of the features (RR1+BW), suggesting that the word features are not particularly useful in this setting.

Although the various ASR systems provide different hypotheses for each decoded segment, the corresponding signal feature vectors are the same and, therefore, the trained regressor could be misled by the co-occurrence of similar feature values with different training labels. To overcome this problem, we trained regression models by using only one random transcription for each segment (RR2). This approach, using both basic and word-based features (RR2+BW), yields the best performance among the RR methods (on average, the WER is reduced by 2.2%). Interestingly, both RR1 and RR2 reach the best performance at combination level $L6$. This behavior, differently from the oracle conditions, indicates that the majority of the words in each confusion set of the ROVER word network is not concentrated on the top "automatically" ranked positions but is more uniformly distributed over all combination levels.

Machine-learned ranking with the basic features (MLR+B), word-based features (MLR+W) and their combination (MLR+BW) always exhibits the largest WER reductions over random ROVER (-2.4% on average). Its results are not only consistently better than RR, but also close to system-level oracle (SysO).

In summary, this preliminary analysis suggests that:

- regardless of the QE method and the features used, QE-informed ROVER outperforms random ROVER with a large margin;
- among all the QE strategies, MLR+BW shows the best performance at most of the levels, and
- though still far from the strongest term of comparison (SegO), its results are competitive with those of the system-level oracle.

As shown in the last column of Table 5.4, the average improvement of the two methods are almost the same (-1.3% vs. -1.1%).

Test. Using the QE models trained on IWSLT2012, we carry out the same set of experiments on the test data. The corresponding results are reported in Table 5.5. Since in IWSLT2013 there are 8 teams who submitted their transcriptions, in this case the combination level varies in the $[L3, L8]$ interval.

Although the overall trend is similar to the one observed on the training data, the improvement margins are smaller. The reason for this difference is the mismatch between training and test data. In our preliminary experiments, the same set of ASR systems (6 systems) was used both in training and test. Here, instead, the ASR systems are different in number and nature (6 systems from IWSLT2012 for training, 8 systems from IWSLT2013 for test). Nevertheless, in many columns of Table 5.5 the WER reduction

Ranking	L3	L4	L5	L6	L7	L8	Avg. Impr
Random	14.6	13.7	13.2	12.8	12.7	12.4	0.0
SysO	12.2	11.7	11.8	11.9	12.1	12.1	-1.3
SegO	10.5	11.0	11.4	11.6	11.7	11.7	-1.9
RR1+B	13.9	13.1	12.6	12.4	12.4	12.3 † •	-0.4
RR1+W	14.0	13.0	12.5	12.2	12.3 •	12.3 † •	-0.5
RR1+BW	14.0	13.0	12.5	12.2	12.3 •	12.3 † •	-0.5
RR2+B	13.8	13.0	12.6	12.4	12.3 •	12.3 † •	-0.5
RR2+W	14.2	13.1	12.7	12.4	12.5 †	12.4 † •	-0.3
RR2+BW	13.7	12.8	12.4	12.2	**12.2** •	**12.2** † •	-0.6
MLR+B	12.9	12.4	12.3	12.1 •	12.3	**12.2** † •	-0.9
MLR+W	**12.4** •	**12.1**	**12.0**	12.0 •	**12.2** •	**12.2** † •	-1.1
MLR+BW	**12.4** •	**12.1**	**12.0** •	**11.9** • ⋆	**12.2** •	**12.2** † •	-1.1

Table 5.5: WER results of different ranking methods on IWSLT2013. $L8$ indicates that all the 8 systems participated in IWSLT2013 are combined. The symbols indicate the statistical significance at the level of 95%. "†": the result is not statistically different from random ROVER; "•": the result is not significantly different from SysO; "⋆": the result is not statistically different from SegO.

is large and significant[6]. The best result (MLR+BW at $L6$), in particular, is not only better than Random ROVER, but also not statistically different from the strongest oracle (SegO). Also on the test set, MLR seems to work better than RR in ranking the input hypotheses. One possible reason is the higher reliability of pairwise comparisons compared to considering numeric scores that reflect independent quality predictions.

Further analysis. So far, we showed that: *i)* we can improve random ROVER using ASR QE for ranking the inputs, and *ii)* with an appropriate QE approach and an efficient set of features, we can approach the strong system-level and segment-level oracles. The scores reported in Table 5.5,

[6]To perform significance test, we exploit the matched-pairs test Gillick and Cox (1989) at a significance level of 95% (*i.e.* $p = 0.05$).

however, only provide a global picture that might hide interesting details such as larger gains in specific conditions that are particularly favorable for QE-based ranking.

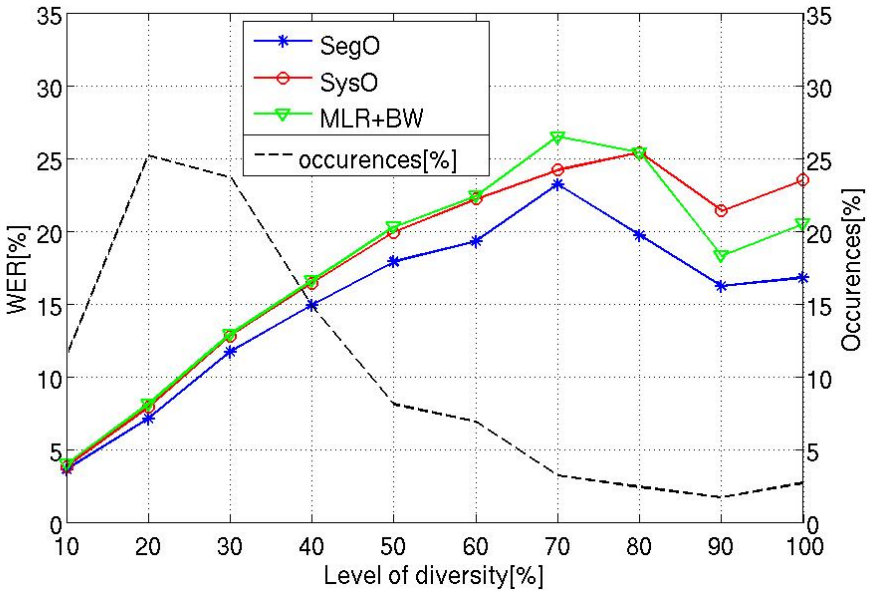


Figure 5.3: WER results achieved on the evaluation set IWSLT2013 as a function of hypothesis diversity ($div = MAX_{WER}[\%] - MIN_{WER}[\%]$)

To better investigate this aspect, in Figure 5.3, we analyse the performance on different groups of segments characterized by different levels of diversity. By diversity, we refer to the level of disagreement between ASR systems. Among the many possible methods to compute diversity measures, here it is computed as the difference between the maximum and minimum WERs among the transcriptions of a given segment. With this definition, we divide the segments into 10 groups (X-axis in Figure 5.3). The first group consists of the segments whose transcriptions diversity is lower than 10% (*i.e.* the WER difference between the best and the worst transcription is less than 10%). In the second group, the diversity is between [10%,20%] and so on. The figure reports the performance of QE-informed ROVER using SegO, SysO and MLR+BW over each diver-

sity group, as well as the proportion of segments belonging to each group (black-dotted line). As it can be seen, most of the segments have low diversity (less than 20%), meaning that the ASR systems return similar transcriptions for them.

It is interesting to note how the relative differences between the three systems are affected by hypothesis diversity. For low values, in which ranking is intuitively more difficult and less accurate, MLR+BW's performance is less reliable and close to SysO. For high values, in which the ranker is able to order the components more precisely, MLR+BW outperforms SysO halving the gap that separates it from SegO. This interesting trend is hidden by the fact that the IWSLT data are characterized by low diversity in the transcriptions (highly-diverse hypotheses can be found for less than 7% of the segments). This, in turn, results in global WER scores where large gains on few segments are smoothed by small gains on many segments.

Though suitable to evaluate our approach, the IWSLT scenario is not the ideal one to fully appreciate its potential. In the next set of experiments we apply QE-informed ROVER to the CHiME-3 data, in which the diversity among the microphone channels is higher.

5.5.4 CHiME-3 task

Data. For the multiple microphones task, we use CHiME-3 data described in §4.4.1. The organizers of the challenge provided two baseline ASR systems employing the Kaldi toolkit (Barker et al., 2015). Both are based on hidden Markov models: one uses the Gaussian mixture model ($*-gmm$) and the other uses deep neural network ($*-dnn$). The former is trained with the Kaldi recipe prepared for the previous CHiME challenges (Barker et al., 2013; Vincent et al., 2013) and the latter is trained with Karel's setup (Vesely et al., 2011) included in the Kaldi toolkit.

In the experiments, we use the *real* noisy subsets. For training the

QE models, *dt05_real* (DT05 henceforth) consisting of 1,640 sentences uttered by four different speakers is utilized. As test data, *et05_real* (ET05 henceforth) consisting of 1,320 sentences uttered by four other speakers is used. In contrast to IWSLT, here no automatic segmentation is needed, since each utterance recording shares the same time segmentation across all microphones.

Tables 5.6 and 5.7 respectively show some statistics about the CHiME-3 data and the WER obtained by the baseline ASR systems over each of the five different channels (microphones). As mentioned also in §5.4, the sixth microphone is not used in the evaluation, as it is located on the back of tablet PC, mainly to capture the background noise.

In distant speech recognition, it is common to combine the signals recorded by a microphone array in order to enhance the signal. As mentioned in §5.2, MVDR and DS are two popular enhancement approaches. We apply both methods to combine the signals from the five microphones and then we transcribe the resulting signals by using the two aforementioned baseline ASR systems. This produces four additional channels whose WERs are reported in Table 5.7. As it can be seen, DS processing significantly outperforms MVDR on both training and test sets. Nevertheless, we keep also the MVDR hypotheses for combination, because they provide complementary solutions whose diversity might improve final results.

Experiments are conducted similarly to the IWSLT task §5.5.3, with the difference that in this case each hypothesis is generated by the two baseline systems (*gmm* and *dnn*) processing each individual microphone audio track ($01, 02, 03, 04, 05, mvdr$ and ds).

Comparing the WER scores reported in Tables 5.3 and 5.7, it is worth noting that the quality of the CHiME-3 transcriptions is globally lower, and on average, training and test data are more distant. As we will discuss

Attributes	DT05 (training)	ET05 (test)
duration (hr)	2h74m	2h33m
# sentences	1,640	1,320
# words	27.1k	21.4k
dict. size	1.6k	1.3k
# speakers	4	4

Table 5.6: Statistics of CHiME-3 task

Channels	DT05	ET05
01-dnn	20.5	32.9
01-gmm	23.4	37.3
02-dnn	20.0	38.8
02-gmm	24.2	40.5
03-dnn	18.8	38.0
03-gmm	21.5	36.5
04-dnn	16.7	32.6
04-gmm	18.7	33.2
05-dnn	16.5	34.4
05-gmm	19.2	34.8
mvdr-gmm	20.3	37.1
mvdr-dnn	17.6	33.1
ds-gmm	12.2	23.1
ds-dnn	10.4	20.5
Avg.	18.6	33.9

Table 5.7: WER results of different recognition channels in CHiME-3 task.

in §5.5, this can explain why: *i)* the performance achieved by standard random ROVER is lower on CHiME-3 data than on IWSLT, *ii)* adding high quality transcriptions to the combination (*e.g.* those obtained from signal enhancement) results in much larger WER reductions on CHiME-3, and *iii)* ranking methods seem to suffer from the presence of a large number of transcriptions with very similar WER scores.

Preliminary analysis. Differently from the IWSLT task, where the transcriptions come from multiple ASR systems, in CHiME-3 the transcriptions come from multiple channels and two ASR systems. Also in this task we first train the QE models on the training set (DT05) using 8-fold cross validation. This partitioning with 8 folds is done intentionally due to avoid speaker and sentence overlaps between training and test folds. As mentioned before, for each utterance there are five signals recorded by the

microphones, plus two enhanced signals. Each signal is transcribed using both GMM and DNN acoustic models. Hence, 14 hypotheses are generated for each segment, and consequently, the combination levels range in the interval [*L3,L14*].

Ranking	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	Avg. Impr.
Random	14.3	13.4	12.7	12.4	12.1	11.9	11.9	11.8	11.8	11.8	11.8	12.2	0.0
SysO	11.0	10.9	10.3	10.4	10.6	10.6	10.9	10.7	10.9	10.9	11.2	11.5	-1.5
SegO	6.8	7.1	7.5	7.8	8.3	8.6	9.2	9.4	10.0	10.4	10.9	11.2	-3.4
RR1+B	10.6	10.3	10.0	9.9	10.0	10.1	10.3	10.4	10.8	11.0	12.0	12.0	-1.7
RR1+W	12.3	11.4	10.9	10.7	10.7	10.7	10.8	11.1	11.3	11.7	12.0	12.1	-1.0
RR1+BW	10.3	10.0	9.7	9.7	9.9	**9.9**	**10.2**	10.4	10.7	**10.0**	11.7	**11.9**	-2.0
MLR+B	10.0	9.7	9.6	9.6	9.8	10.0	**10.2**	10.4	**10.5**	10.7	**11.3**	**11.9**	-2.0
MLR+W	10.7	10.5	10.3	10.3	10.3	10.5	10.7	11.0	11.3	11.5	11.6	12.0	-1.4
MLR+BW	**9.8**	**9.5**	**9.5**	**9.5**	**9.7**	**9.9**	**10.2**	**10.3**	**10.5**	10.8	11.5	**11.9**	-2.1

Table 5.8: WER results of different ranking methods on the CHiME-3 training set (DT05) using 8-fold CV. $L14$ indicates that all the 14 systems, i.e. 7 signals (5 from microphones + 2 enhanced), each transcribed by 2 ASR systems (GMM and DNN) are participated in the combination.

Differently from IWSLT, in this task we do not take into account the RR2 approach, which uses only one of the hypotheses (among all available transcription channels) to train the regression models. Because in this task, the microphones are positioned in different places around the tablet PC and each of them captures the signal from a different angel. This makes the signal-based features different from one hypothesis to the other.

The results reported in Table 5.8 show that, similar to the IWSLT task, the best performance is achieved in most of the cases by MLR+BW (with an average WER reduction of 2.1% over random ROVER). Unlike IWSLT, most of our results outperform the SysO oracle, which always uses the enhanced channels as the first input. The best WERs are usually obtained

at the lower levels (*i.e.* 9.5% at $L4$, $L5$ and $L6$). One explanation can be the large gap between the best channels ($ds - gmm$ and $ds - dnn$, respectively obtaining 12.2% and 10.4% WER) and the other channels (obtaining from 16.5% to 24.2% WER). By adding more hypotheses, worse transcriptions are considered in the combination and, consequently, majority voting makes more mistakes. At $L13$ and $L14$, indeed, our results remain slightly worse than SysO.

An interesting achievement with this approach is the significant improvement of hypothesis-level combination with respect to the signal-level combination. The top result obtained by MLR+BW (9.5%) is indeed better than the best one reported in Table 5.7 for signal-level combination (10.4% with $ds - dnn$). In contrast with previous works suggesting that hypothesis-level combination does not yield any significant improvement on top of signal-level combination (Stolcke, 2011), our results show that better final results can be obtained if the input components are ordered accurately.

Test. Table 5.9 includes the results achieved by QE models trained on *DT05* data and then used to predict the quality of the *ET05* transcriptions. The observations of our preliminary analysis seem to be confirmed. Although trained and tested on different data, QE-informed ROVER consistently outperforms random ROVER and the SysO oracle at all levels except for $L14$, where the WER achieved by our best system is not statistically significant. This supports our hypothesis that, if the diversity among the components is high enough, then the proposed segment-level QE-informed ROVER can lead to better results than the system-level oracle. SegO performance is still significantly better at all levels, especially the lower ones where the transcriptions obtained from the best enhanced channels are hard to be improved with hypotheses coming from the other

Ranking	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	Avg. Impr.
Random	28.7	27.4	27.2	26.8	26.5	26.4	26.2	26.2	26.1	26.1	26.2	26.3	0.0
SysO	22.7	22.0	21.6	21.1	21.9	22.0	24.0	24.1	24.4	24.9	25.5	25.8	-3.3
SegO	14.8	15.4	16.2	17.0	18.1	18.7	19.7	20.3	21.1	21.9	22.8	23.6	-7.4
RR1+B	20.4	**19.5**	19.5	20.1	20.2	20.6	21.1	21.7	22.2	**22.9**	24.2	25.8†•	-5.1
RR1+W	22.2•	21.1	20.3	20.3	20.5	21.0	21.3	22.0	22.9	24.0	24.9	26.0†•	-4.5
RR1+BW	20.0	**19.5**	**19.1**	**19.5**	**19.7**	**20.3**	**20.7**	**21.4**	**22.1**	**22.9**	**23.9**	25.8†•	-5.4
MLR+B	20.4	19.9	19.6	20.0	20.3	20.6	21.2	21.5	22.5	23.3	24.9	25.8†•	-5.0
MLR+W	22.4•	21.4	20.8	20.7	21.1	21.5	22.0	22.6	23.2	24.0	25.2	25.9†•	-4.1
MLR+BW	**19.8**	**19.5**	19.5	19.7	20.2	20.4	20.9	21.5	22.2	23.4	24.9	**25.7**•	-5.2

Table 5.9: WER results of different ranking methods on the CHiME-3 test set *ET05*. The symbols indicate the statistical significance test at the level of 95%. "†": the result is not significantly different from random ROVER; "•": the result is not significantly different from SysO; "⋆": the result is not significantly different from SegO.

channels.

It is important to remark that QE-informed ROVER also significantly improves the performance of the enhanced channels when combining less than eight transcriptions. In fact, as shown in Table 5.7, the enhanced $ds - dnn$ channel achieves 20.5% WER on ET05, while RR1+BW reduces the error down to 19.1% at $L5$. As mentioned before, probably due to the high performance difference between the enhanced and raw channels, we do not observe the same gain at higher levels.

Differently from IWSLT, where there is an increment of 2% WER when moving from the preliminary analysis to the test results, in CHiME-3 we observe a larger degradation (10% WER on average). This is not surprising by looking at the differences in WER between training and test data in the two tasks (Tables 5.3 and 5.7), which indicate that CHiME-3 data is more heterogeneous. However, though working in a more complex scenario, QE-informed ROVER still achieves competitive results.

About learning algorithms, in most of the cases the best performance on CHiME-3 is obtained by RR1. This is in contrast with the IWSLT results, where MLR always performed better than RR. The main reason for this is related to the fact that in CHiME-3, apart from the enhanced channels, the other channels are quite similar in performance and, overall, they generate transcriptions of low quality with similar or equal WERs (see Table 5.7). For MLR, this results in a large number of ties when computing the pairwise comparisons. The impact of ties on MLR performance is addressed in §5.6.1.

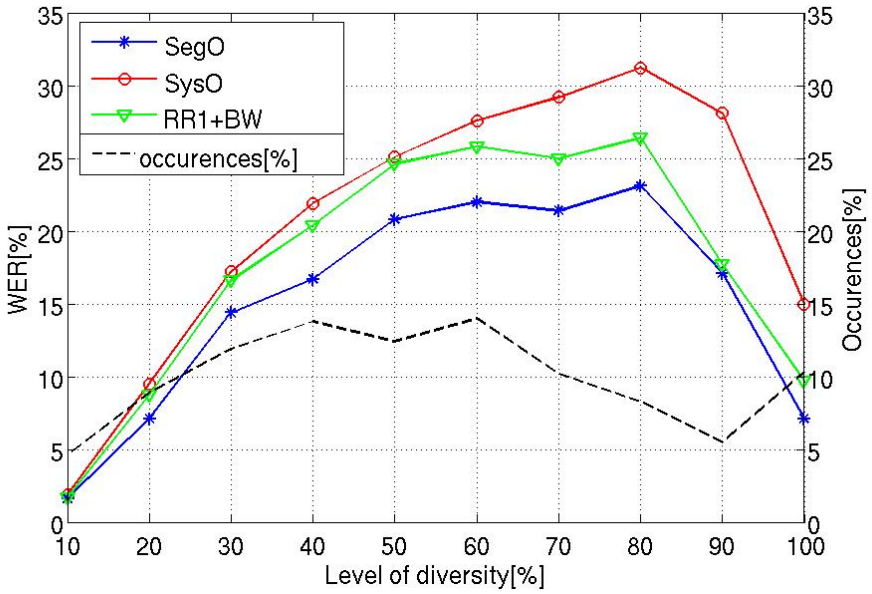


Figure 5.4: CHiME-3

Figure 5.5: WER results achieved on the evaluation set *ET05* as a function of hypothesis diversity ($div = MAX_{WER}[\%] - MIN_{WER}[\%]$)

Further analysis. Figure 5.5 shows the performance analysis with regard to the diversity groups in the CHiME-3 task. Comparing the curves in Figures 5.3 and 5.5, we observe that the distribution of the CHiME-3 transcriptions with respect to the different levels of diversity (black-dotted lines) is more

uniform than in IWSLT. This is mostly due to the presence of the enhanced channels that perform significantly better than the raw ones, hence they enlarge the gap between the best and the worst transcriptions. In terms of performance, this uniform distribution allows QE-informed ROVER to outperform SysO also for small values of diversity (30%) in CHiME-3. Increasing the level of diversity, our method significantly gains over SysO and also approaches SegO at the diversity value of 90%.

5.6 Discussion

We have shown that:

- QE-informed ROVER is able to achieve better performance than standard ROVER on two very different datasets;
- our approach also outperforms the system-level oracle in the CHiME-3 task, and it obtains competitive results with the enhanced channels;
- the extent of our gains depends on the level of diversity and accuracy of the transcriptions, and
- the lowest WER scores are obtained combining a limited number of transcriptions.

Despite these positive results, our analysis still leaves two important questions open:

1. Why machine-learned ranking (MLR+BW) shows better performance on IWSLT, whereas in CHiME-3 ranking by regression (RR1+BW) generally works better?
2. Considering that the best performance is always obtained at a task-specific level of combination, how can we predict this level?

In this section we address these two practical issues, respectively by:

1. analyzing the impact of tied ranks (transcriptions with identical WERs) in the CHiME-3 training data and

2. exploring a classification method to identify the optimum combination level.

5.6.1 Tied ranks

In the previous section we observed that the highest results on IWSLT and CHiME-3 data are respectively obtained by MLR and RR. This raises the following practical issue: *how to find a unique, best performing strategy, suitable for the general case?*

One reason for the observed difference can be found in the way the QE models are trained. In RR they are trained to separately predict the WER of each transcription, while in MLR they are trained to predict relative ranks through pairwise comparisons. Training sets characterized by a large number of ties (transcriptions of the same segment with identical WERs but different quality) may influence the performance of MLR more than RR.[7]

Table 5.10 reports the average percentage of ties (similar or identical hypotheses) for each dataset, showing that IWSLT has less ties than CHiME-3 both in the training and test sets. This seems to contradict the plots of Figure 5.5, which indicate a higher diversity in the CHiME-3 data (compared to IWSLT, instances are more evenly distributed across all diversity levels). Such diversity, however, is only due to the presence of the enhanced channels that are far better than the raw ones. Indeed, in CHiME-3 we observe a lot of ties among the raw channels (i.e. the transcriptions gen-

[7]This problem is widely explored in the information retrieval field, where ties are either arbitrarily broken (Fürnkranz and Hüllermeier, 2003) or managed with *ad-hoc* strategies (Zhou et al., 2008).

erated by the non-enhanced signals). Moreover, in CHiME-3, the training set includes 8% more ties than the test set. This suggests that the presence of a large number of ties in the training set of CHiME-3 can be critical and, in turn, it can determine the lower results achieved by MLR.

	IWSLT	CHiME-3
Train	38.4%	58.9%
Test	40.5%	50.6%

Table 5.10: Percentage of ties (similar or identical hypotheses) in each dataset.

To validate this hypothesis, in the following experiments we break the ties in the training set by looking at the global performance of each system/channel on the same data.[8] In particular, if two systems, A and B, achieve the same WER on a given training segment, and system A has shown to perform better than system B on the whole training set, then the hypothesis suggested by A will be prioritized when breaking the ties.

IWSLT	L3	L4	L5	L6	L7	L8		-	-	-	-	-	Avg. Impr.
RR2+BW	13.7	12.8	12.4	12.2	12.2•	**12.2**†•		-	-	-	-	-	-0.6
MLR+BW	12.4•	12.1	**12.0**•	**11.9**•⋆	12.2•	**12.2**†•		-	-	-	-	-	-1.1
MLR+BW +Untied	**12.3**	**11.9**•	**12.0**•	**11.9**•⋆	**12.1**•	**12.2**†•		-	-	-	-	-	-1.2
CHiME-3	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	Avg. Impr.
RR1+BW	20.0	19.5	**19.1**	**19.5**	**19.7**	**20.3**	**20.7**	21.4	22.1	22.9	23.9	25.8†•	-5.4
MLR+BW	19.8	19.5	19.5	19.7	20.2	20.4	20.9	21.5	22.2	23.4	24.9	25.7•	-5.2
MLR+BW +Untied	**19.3**	**18.6**	19.2	**19.5**	20.0	**20.3**	20.8	**21.2**	**21.8**	**22.5**	**23.2**	**24.0**⋆	-5.8

Table 5.11: WER results when the ties are broken using prior knowledge.

[8]This information is similar to the prior knowledge that the SysO oracle acquires on the training data and exploits to rank the test hypotheses. Here, however, we fairly operate only on the training set.

Table 5.11 shows how performance varies if the ties in the training data are broken using such prior knowledge. In IWSLT, the MLR approach trained on untied ranks (MLR+BW+Untied) achieves minor improvements. This is due to the fact that: *i)* the number of ties is not so high to represent a critical issue, as we saw in Table 5.10, and *ii)* the WER difference between the different systems and levels are such minimal that reduces the room for improvement. In CHiME-3, instead, the model learned from the untied training set (MLR+BW+Untied) significantly outperforms MLR+BW at all levels of combination and also outperforms RR1+BW at most of the levels. The considerable WER result of 18.6% obtained by this method at $L4$, which also corresponds to a relative improvement of 9.6% over the best enhanced channel, represents our best result on CHiME-3. This confirms the validity of our intuition about the negative impact of tied ranks on MLR performance. The largest improvement is obtained when combining all the transcriptions ($L14$). This is not surprising because this is the condition where more ties occur. Interestingly, at this level, the results of QE-informed ROVER improve up to the point that they are no longer statistically different from the strong segment level oracle (see Table 5.9).

Similarly to what we observed in the previous experiments, the best results are obtained at low levels of combination. This calls for solutions to automatically find (or at least approximate) the optimum level, which is the problem investigated in the next section.

5.6.2 Optimum level of combination

Table 5.11 shows that, after breaking the ties, the optimum level of combination in IWSLT (8 components) is either $L4$ or $L6$, while in CHiME-3 (14 components) it is $L4$. A *post-hoc* comparison of the results achieved by each level on the test set, however, is only made possible by the availability

of references to compute final WER scores. This observation raises a new practical issue: *how can we predict the optimum level of combination in a real condition in which we have no access to the reference transcripts?* The problem of finding a dataset-specific stopping criterion, as an alternative to the simple and risky "take-all" strategy, is well motivated. Although it seems less important for IWSLT, where final WER scores are quite similar for all the levels, a method to avoid entering harmful inputs into the ROVER combination can significantly change the results in CHiME-3.

Several sub-optimal solutions can be applied to address this problem. The simplest one is the random choice, which is acceptable in situations where hypothesis quality is homogeneous (IWSLT) but, as we will show below, it can be inadequate when the variability is higher (CHiME-3). Another option is to determine the optimal level in the training set and apply it to the test data. This, however, would not be possible when the number of components varies between training and test, as in the case of the IWSLT task.

A better solution is suggested by the findings in (Audhkhasi et al., 2014) that demonstrates the strong dependency between the performance of ASR system combination methods and the diversity and quality of the components. Following this intuition, we explore a classification-based approach, in which a binary classifier is trained to learn wether a combination level is appropriate (labeled as *1*) or not (labeled as *0*). Based on the predictions of the classifier, for each given segment we select the appropriate level. In case of finding multiple optimum levels, we rely on the confidence score of the classifier.

To prepare the training data, for each segment we first run QE-informed ROVER at all the levels. Then, label *1* is assigned to the level(s) with the lowest WER. It is probable that for some segments, different combination levels may result in the same lowest WER score. When this happens, multi-

ple levels are labeled as *1*, thus resulting in skewed label distributions. This is the case of IWSLT data, especially the test set, in which more levels have identical scores compared to CHiME-3 (the percentage of positive examples in the two test sets is respectively 77.0 and 57.3). Looking at the results in Table 5.11 this is not surprising, since the WER difference between the best and the worst levels is minimal (0.4% for MLR+BW+Untied) compared to CHiME-3 (5.4%). Moreover, looking at the distributions in Figures 5.3 and 5.5, we notice that the majority of the IWSLT segments have transcriptions with small diversity. These considerations indicate a higher probability to find transcriptions of similar quality in the IWSLT data and, in turn, a more skewed label distribution when training our classifier.

Features

The features for this classification task are extracted by using the confusion networks generated by ROVER at each combination level. In particular, we compute:

1. Overall diversity of the level, computed using the theorem in (Audhkhasi et al., 2014);
2. levenshtein distance between the first and the last hypotheses;
3. avg. Levenshtein distance between the first hypothesis and the others;
4. avg. Levenshtein distance between each hypothesis and the next one;
5. avg. Levenshtein distance between each hypothesis and the final combination;
6. mean predicted WER obtained by RR methods among the hypotheses.
7. minimum predicted WER obtained by RR methods among the hypotheses.

8. maximum predicted WER obtained by RR methods among the hypotheses.

Except for the first feature, the others are quite simple to compute. The theorem of Audhkhasi et al. (2014) defines diversity as the average of the approximate WERs of the individual systems from the ROVERs prediction. Based on this definition, the diversity of a confusion network generated by ROVER is computed by:

$$Diversity = \frac{1}{I \times M} \sum_{i=1}^{I} \sum_{m=1}^{M} \frac{1}{2} \|h_i^{avg} - h_i^*\|_2^2$$

$$\text{where,} \quad h_i^{avg} = \frac{1}{M} \sum_{m=1}^{M} h_i^m$$

$$\text{where,} \quad h_i^m = \alpha w_i^m + (1 - \alpha) s_i^m$$

In this equation, I is the total number of bins (slots) in the confusion network and M is the number of hypotheses to be combined. h_i^{avg} is the average of h_i^m vectors in i-th bin and h_i^* represents the ROVER's output and it is obtained by setting the maximum element of h_i^{avg} as 1 and the rest 0. h_i^m is a counter-vector for the word w_i^m appeared in the m-th hypothesis. w_i^m is indeed a one-hot vector with the same dimension as the vocabulary size and s_i^m represents the confidence score of this word. α is a coefficient that has to be tuned. Since in our task, the confidence scores are not available, we set $\alpha = 1$.

Classifiers

As classification algorithms, we experimented with support vector machine (SVM) (Mammone et al., 2009), random forest (Breiman, 2001) and BayesNet (Friedman et al., 1997). Balanced accuracy, which is particularly

appropriate in case of skewed distributions, is used as evaluation metric to optimize hyper-parameters and select the best classification algorithm. The results reported in Table 5.12 are computed in 5-fold cross validation on the training set and they show that BayesNet outperforms the other classifiers and, by a large margin, also the 50.0% balanced accuracy reachable with a baseline majority voting classifier. One possible explanation for the success of BayesNet could be in its higher capability to work with a limited number of features (i.e. 8 features in these experiments) and a large number of instances (4496[9] for IWSLT and 19680[10] for CHiME-3). Comparing the performance obtained on the two datasets, the fact that CHiME-3 results are better than the IWSLT ones suggests that the more skewed distribution of the IWSLT labels penalizes the classifier.

Task	**IWSLT**	**CHiME-3**
Classifier	Balanced Acc. IWSLT2012	Balanced Acc. DT05
SVM	52.8	66.1
Random Forest	58.7	71.5
BayesNet	**65.5**	**72.0**

Table 5.12: Performance of different binary classifiers used to find the optimum level of combination.

In light of its higher balanced accuracy, the BayesNet classifier has been selected to predict the best level of combination for each segment in the test data. On both tasks the results outperform those of the sub-optimal strategy based on random selection of the best combination level. While on IWSLT the WER reduction is unsurprisingly minimal (from 12.1% using random selection to 12.0% using BayesNet), on CHiME-3 the gain is much

[9] $1,124 \times 4 = 4496$, as there are 1,124 utterances in IWSLT2012 that can be combined in 4 levels $[L3, L4, L5, L6]$.

[10] $1,640 \times 12 = 19,680$, as there are 1,640 utterances in DT05 that can be combined in 12 levels $[L3, .., L14]$.

larger (from 20.9% using random selection to 18.8% using BayesNet).[11] More interestingly, besides outperforming random selection, our classifier provides predictions that closely approximate the performance of the best levels shown in Table 5.11 (11.9% for IWSLT and 18.6% for CHiME-3).

Overall, these results indicate that:

- The optimal combination does not only depend on the quality of the hypotheses and the reliability of their ranking but also on their diversity (this is in line with the discussion in Section §5.5.4 and with the results reported in (Audhkhasi et al., 2014));
- The relative importance and contribution of these aspects can be learned from data;
- The resulting models can effectively support our QE-informed ROVER approach in real operating conditions where reference transcripts are not available.

5.7 Summary

QE-informed ROVER, as an approach to perform ROVER system combination on the 1-best hypotheses coming from multiple transcription channels was presented in this Chapter. We used ASR QE to address three issues about the classic ROVER, including *i)* the order of the entering hypotheses; *ii)* low granularity level and *iii)* dependency on confidence score and ASR decoder features. To this aim, we applied ASR QE to automatically rank the inputs at segment-level (the first and second limitations). Moreover, we used only the black-box features that do not need the ASR decoder information (the third issue). By exploiting QE-informed ROVER,

[11] The baseline sub-optimal scores are obtained by averaging the results obtained from five iterations of the random selection process.

we observed consistent, positive improvements in two very different combination scenarios. The first scenario, IWSLT involves multiple ASR systems and close-talk English TED talks that are transcribed by 8 unknown ASR engines. The second scenario involves noisy signals recorded by multiple microphones to be combined. On both experiments, QE-informed ROVER outperformed the classic ROVER significantly and it closely approached the oracle results.

Chapter 6

Conclusion

This PhD explored automatic quality estimation for the ASR outputs. The study was first motivated by the fact the actual

- ASR evaluation is quite expensive in terms of human effort;
- manual reference is not always available and
- the ASR decoder information is not always accessible.

We described the principles of ASR QE in Chapter 3. We defined ASR QE as an automatic quality evaluation system which differs from confidence measuring and confidence estimation studied during the last decades. The main difference between confidence estimation and quality estimation is indeed the lack of ASR decoder information when we talk about ASR QE. A wide range of sentence-level and word-level features were presented in Chapter 3. We categorized all these features into four groups: signal, textual, hybrid and word-level features. The efficacy and complementarity of these features were confirmed in several experiments.

Then we introduced different learning mechanisms for two distinct scenarios. The first scenario, single hypothesis, involves only one transcription hypothesis per utterance. This scenario was explored as word error rate (WER) prediction task. We showed that extremely randomized tree (XRT)

fits very well as a regression model for WER prediction. The second scenario, multiple hypotheses, involves several transcription hypotheses per utterance. In this scenario the goal is to rank the hypotheses according to their quality. For this scenario, we showed that machine-learned ranking (MLR) approache outperform the regression-based one. The advantage of MLR is its ability to perform pairwise comparison between several hypotheses. Among all possible MLR models, we found that random forest (RF) presented the most reliable rankings.

Moving from theory to practice, in the subsequent chapters 4 and 5, we encountered different applications of ASR QE yielding significant WER reduction in the final output.

In the first application, in Chapter 4, we made use of single hypothesis ASR QE to perform deep neural network (DNN) acoustic model adaptation. DNN acoustic models, because of their huge number of parameters cannot be easily adapted with the common adaptation approaches such as MAP and MLLR. To this aim, we use a method based on Kullback-Leiber divergence regularization. In this method, the quality of the adaptation data and also the weight which is given to each adaptation utterance plays a critical role. We proposed to exploit single hypothesis ASR QE approaches to predict the quality of each hypothesis and remove the bad ones from the adaptation data. Moreover, we observed that the predicted values provide a reliable weight to each utterance for performing soft adaptation. In soft adaptation the weight of each adaptation data varies from one instance to the other. Our results showed over 1.8% WER reduction with regard to the strong baseline provided in CHiME-3 challenge.

The second application, in Chapter 5, exploits multiple hypotheses ASR QE for improving system combination. ROVER (random output voting error rate), as one of the most popular ASR hypotheses combiners, is very sensitive to the order of the entering hypotheses. Based on this observa-

tion, we proposed to apply ASR QE to order the hypotheses before running ROVER. On two distinct dataset with two different speech recognition circumstances: one with single closed talk microphone transcribed by multiple ASR systems and the other with multiple distant microphones transcribed by two ASR systems, we showed significant WER reduction achieved by QE-informed ROVER. In the former, we achieved 1.6% absolute WER reduction wrt to the best individual channel and 0.5% absolute reduction wrt to the state-of-the-art combination. In the latter, we showed 14.0% absolute WER reduction wrt to the best individual channel and 1.9% absolute WER reduction wrt to the state-of-the-art combination. In both scenarios, the diversity and complementarity of the hypotheses components played a critical role in the performance of ASR QE and also in the result of the combination procedure.

Moreover, the problem of the tied ranks was addressed in these studies. We showed that breaking the ties based on the overall performance of the transcription channels can be a simple though effective solution. We also investigated the problem of finding the optimum level of combination. The solution was to simply use a binary classifier showing if a combination level is appropriate or not. The empirical results showed that for this classification task BayesNet worked very well. The main features were based on the diversity among the hypotheses in a level. Features such as the mean and standard deviation of the predicted WER values; the difference between the first component with the last one and like on.

6.1 Future work

This PhD research has opened a wide range of possibilities for improving the real-world ASR applications using automatic quality estimation systems.

First of all, the ASR QE by itself has a large space to be improved from different perspectives. In this PhD, we focused mainly on the sentence-level QE. Whereas, working on the phone-level, word-level or even document-level can be done in future. Moreover, extending the features to signal processing-oriented technologies, linguistics information and like on, is another trend that we will follow in future.

ASR QE may lead to incremental learning of the acoustic models. One of the future works, that is indeed in progress, is to use ASR QE for selecting the more suitable training data. In ideal case, an ASR system can be retrained on a portion of the automatically transcribed data that shows higher quality. From this perspective, the ASR system can transcribe the new unseen data, select the high quality portion and retrain itself on that portion.

Bibliography

Kacem Abida, Fakhri Karray, and Wafa Abida. cROVER: Improving ROVER using Automatic Error Detection. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1753–1756, Prague, Czech Republic, 2011.

Victor Abrash, Sankar Ananth Franco, Horacio, and Michael Cohen. Connectionist Speaker Normalization and Adaptation. In *Proc. of he International Speech Communication Association (Interspeech)*, pages 2183–2186, Madrid, Spain, 1995.

Xavier Anguera, Chuck Wooters, and Javier Hernando. Acoustic Beamforming for Speaker Diarization of Meetings. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2011–2022, 2007. ISSN 1558-7916.

X. Aubert, C. Dugast, H. Ney, and V. Steinbiss. Large vocabulary continuous speech recognition of wall street journal data. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP)*, volume ii, pages II/129–II/132 vol.2, 1994. doi: 10.1109/ICASSP.1994.389702.

Kartik Audhkhasi, Andreas M Zavou, Panayiotis G Georgiou, and Shrikanth S Narayanan. Theoretical Analysis of Diversity in an Ensemble of Automatic Speech Recognition Systems. *IEEE/ACM Transactions on Audio, Speech & Language Processing*, 22(3):711–726, 2014.

Nguyen Bach, Fei Huang, and Yaser Al-Onaizan. Goodness: a Method for Measuring Machine Translation Confidence. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 211–219, Portland, Oregon, USA, 2011.

Jon Barker, Emmanuel Vincent, Ning Ma, Heidi Christensen, and Phil Green. The PASCAL CHiME Speech Separation and Recognition Challenge. *Computer Speech & Language*, 27(3):621–633, 2013.

Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe. The third 'CHiME' Speech Separation and Recognition Challenge: Dataset, Task and Baselines. In *Proc. of the 15th IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–9, Scottsdale, Arizona, USA, 2015.

Sara Basson, Dimitri Kanevsky, and Emmanuel Yashchin. Collaboration of multiple automatic speech recognition (asr) systems, 2003. US Patent App. 10/058,143.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. *Noise Reduction in Speech Processing*, volume 2. Springer Science & Business Media, 2009. ISBN 978-3-642-00295-3.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.*, 3:1137–1155, 2003. ISSN 1532-4435.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

Mikael Boden. A guide to recurrent neural networks and backpropagation. *The Dallas project, SICS technical report*, 2002.

Paul Boersma and David Weenink. Praat: Doing Phonetics by (Version 4.3.01). *Retrieved from http://www.praat.org/*, 2005.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. Findings of the 2016 Conference on Machine Translation (WMT16). In *Proceedings of the First Conference on Machine Translation (WMT)*, volume 2, pages 131–198, 2016. ISBN 978-1-945626-10-4.

Ondvrej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proc. of the 10th Workshop on Statistical Machine Translation (WMT)*, pages 1–46, Lisbon, Portugal, 2015.

Fethi Bougares, Del'eglise, Est'eve Paul, Yannick, and Mickael Rouvier. LIUM ASR System for Etape French Evaluation Campaign: Experiments on System Combination using Open-source Recognizers. In *Proc. of the 16th International Conference on Text, Speech, and Dialogue*, pages 319–326, Pilsen, Czech Republic, 2013.

Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to Rank using Gradient De-

scent. In *Proc. of the 22nd international conference on Machine learning*, pages 89–96. ACM, 2005.

Jos'e G. C. de Souza, Marco Turchi, and Matteo Negri. Machine Translation Quality Estimation Across Domains. In *Proc. of the 25th International Conference on Computational Linguistics (COLING)*, pages 409–420, Dublin, Ireland, 2014.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proc. of the Seventh Workshop on Statistical Machine Translation (WMT'12)*, pages 10–51, Montréal, Canada, 2012.

Jos Guilherme Camargo de Souza. *Adaptive Quality Estimation for Machine Translation and Automatic Speech Recognition.* PhD thesis, University of Trento, May 2016.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to Rank: from Pairwise Approach to Listwise Approach. In *Proc. of the 24th International Conference on Machine Learning (ICML)*, pages 129–136, Corvalis, Oregon, USA, 2007.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. Report on the 10th IWSLT Evaluation Campaign. In *Proc. of the 10th International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

Stanley F Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4): 359–393, 1999.

Stephan Clemencon, Marine Depecker, and Nicolas Vayatis. Ranking Forests. *Machine Learning Research*, 14(1):39–73, 2013.

V Dang. RankLib. https://sourceforge.net/p/lemur/wiki/RankLib/, 2013.

Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28 (4):357–366, 1980.

Renato De Mori and Fabio Brugnara. HMM Methods in Speech Recognition. http://www.cslu.ogi.edu/HLTsurvey/ch1node7.html, 1996.

Jos'e G. C. de Souza, Christian Buck, Marco Turchi, and Matteo Negri. FBK-UEdin Participation to the WMT13 Quality Estimation Shared Task. In *Proc. of the 8th Workshop on Statistical Machine Translation (WMT)*, pages 352–358, Sofia, Bulgaria, 2013.

Jos'e G. C. de Souza, Jes'us Gonz'alez-Rubio, Christian Buck, Marco Turchi, and Matteo Negri. FBK-UPV-UEdin Participation in the WMT14 Quality Estimation Shared-task. In *Proc. of the 9th Workshop on Statistical Machine Translation (WMT)*, pages 322–328, Baltimore, MD, USA, 2014.

Jose G. C. de Souza, Hamed Zamani, Matteo Negri, Marco Turchi, and Daniele Falavigna. Multitask Learning for Adaptive Quality Estimation of Automatically Transcribed Utterances. In *Proc. of NAACL*, pages 714–724, Denver, Colorado, 2015.

Gunnar Evermann and PC Woodland. Posterior Probability Decoding, Confidence Estimation and System Combination. In *Proc. of NIST Speech Transcription Workshop*, volume 27, College Park, MD, USA, 2000.

Daniele Falavigna, Marco Matassoni, Shahab Jalalvand, Matteo Negri, and Marco Turchi. DNN adaptation by automatic quality estimation of ASR hypotheses. *Computer Speech & Language*, 2016. ISSN 0885-2308. doi: http://dx.doi.org/10.1016/j.csl.2016.11.002.

Marcello Federico, Luisa Bentivogli, Michael Paul, and Sebastian Stüker. Overview of the IWSLT 2012 Evaluation Campaign. In *Proc. of the 9th International Workshop on Spoken Language Translation (IWSLT)*, pages 11–27, Hong Kong, 2012.

Jonathan G Fiscus. A Post-processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER). In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 347–354, Santa Barbara, CA, USA, 1997. IEEE. ISBN 0-7803-3698-4.

Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian Network Classifiers. *Machine Learning*, 29(2-3):131–163, 1997. ISSN 0885-6125. doi: 10.1023/A:1007465528199.

Johannes Fürnkranz and Eyke Hüllermeier. Pairwise Preference Learning and Ranking. In *Proc. of the 14th European Conference on Machine Learning (ECML)*, pages 145–156, Cavtat, Croatia, 2003.

Sadaoki Furui. Cepstral Analysis Technique for Automatic Speaker Verification. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(2):254–272, 1981.

M. J. F. Gales. Maximum Likelihood Linear Transformations for HMM-based Speech Recognition. *Computer Speech and Language*, 12:75–98, 1998.

Simona Gandrabur, George Foster, and Guy Lapalme. Confidence Estimation for NLP Applications. *ACM Transactions on Speech and Language Processing*, 2006.

Sri Garimella, Arindam Mandal, Nikko Strom, Bjorn Hoffmeister, Spyros Matsoukas, Sree Hari, and Krishnan Parthasarathi. Robust i-vector based Adaptation of DNN Acoustic Model for Speech Recognition. In *Proc. of the 9th International Conference on Spoken Language Processing (INTERSPEECH)*, Dresden, Germany, 2015.

Roberto Gemello, Franco Mana, Stefano Scanzio, Pietro Laface, and Renato De Mori. Linear Hidden Transformations for Adaptation of Hybrid ANN/HMM Models. *Speech Communication*, 49(10):827–835, 2007.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely Randomized Trees. *Machine learning*, 63(1):3–42, 2006.

Laurence Gillick and Stephen J Cox. Some Statistical Issues in the Comparison of Speech Recognition Algorithms. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 532–535, Glasgow, Scotland, 1989.

Sharon Goldwater, Dan Jurafsky, and Christopher D Manning. Which Words Are Hard to Recognize? Prosodic, Lexical, and Disfluency Factors that Increase Speech Recognition Error Rates. *Speech Communication*, 52(3):181–200, 2010.

C. Gollan and M. Bacchiani. Confidence Scores for Acoustic Model Adaptation. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4289–4292, Las Vegas, Nevada, USA, 2008.

Cristina Guerrero and Maurizio Omologo. Exploiting Inter-microphone Agreement for Hypothesis Combination in Distant Speech Recognition. In *Proc. of the 22nd European Signal Processing Conference (EUSIPCO)*, pages 2385–2389, Lisbon, Portugal, 2014.

LI Hang. A Short Introduction to Learning to Rank. *IEICE Transactions on Information and Systems*, 94(10):1854–1862, 2011.

Hynek Hermansky. Perceptual linear predictive (PLP) analysis of speech. *the Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.

Dustin Hillard, Bjoern Hoffmeister, Mari Ostendorf, Ralf Schlueter, and Hermann Ney. iROVER: Improving System Combination with Classification. In *Proc. of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 65–68, Rochester, New York, 2007.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine*, 9 (3):82–97, 2012.

Björn Hoffmeister, Tobias Klein, Ralf Schlüter, and Hermann Ney. Frame Based System Combination and a Comparison with Weighted ROVER and CNC. In *Proc. of the 9th International Conference on Spoken Language Processing (INTERSPEECH)*, pages 537–540, Pittsburgh, PA, USA, 2006.

Björn Hoffmeister, Dustin Hillard, Stefan Hahn, Ralf Schlüter, Mari Ostendorf, and Hermann Ney. Cross-Site and Intra-Site ASR System Combination: Comparisons on Lattice and 1-Best Methods. In *Proc. of the*

IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pages 1145–1148, Honolulu, HI, USA, 2007.

Takaaki Hori, Zhuo Chen, Hakan Erdogan, John R. Hershey, Jonathan Le Roux, Vikramjit Mitra, and Shinji Watanabe. The MERL/SRI System for the 3rd CHiME Challenge using Beamforming, Robust Feature Extraction, and Advanced Speech Recognition. In *IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 475–481, 2015.

Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001. ISBN 0130226165.

Zhiheng Huang, Geoffrey Zweig, and Benoit Dumoulin. Cache based recurrent neural network language model inference for first pass speech recognition. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6354–6358. IEEE, 2014.

Shahab Jalalvand. Improving Language Model Adaptation using Automatic Data Selection and Neural Network. In *Recent Advances in Natural Language Processing*, pages 86–92, Hissar, Bulgaria, September 2013.

Shahab Jalalvand and Daniele Falavigna. Parameter Optimization for Iterative Confusion Network Decoding in Weather-Domain Speech Recognition. In *Proc. of the 10th International Workshop on Spoken Language Translation (IWSLT)*, pages 333–337, Heidelberg, Germany, 2013.

Shahab Jalalvand and Daniele Falavigna. Direct Word Graph Rescoring Using A* Search and RNNLM. In *Proc. of the 15th Annual Conference of*

the International Speech Communication Association (INTERSPEECH), pages 2630–2634, Singapore, 2014.

Shahab Jalalvand and Daniele Falavigna. Stacked Auto-Encoder for ASR Error Detection and Word Error Rate Prediction. In *Proc. of the 16th Annual Conference of the International Speech Communication Association (INTERPSEECH)*, pages 2142–2146, Dresden, Germany, 2015.

Shahab Jalalvand, Ahmad Akbari, and Babak Nasersharif. A Classifier Combination Approach for Farsi Accents Recognition. In *20th Iranian Conference on Electrical Engineering (ICEE)*, pages 716–720, May 2012. doi: 10.1109/IranianCEE.2012.6292447.

Shahab Jalalvand, Daniele Falavigna, Marco Matassoni, Piergiorgio Svaizer, and Maurizio Omologo. Boosted Acoustic Model Learning and Hypotheses Rescoring on the CHiME-3 Task. In *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 409–415, Scottsdale, Arizona, USA, 2015a.

Shahab Jalalvand, Matteo Negri, Daniele Falavigna, and Marco Turchi. Driving ROVER With Segment-based ASR Quality Estimation. In *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1095–1105, Beijing, China, 2015b.

Shahab Jalalvand, Matteo Negri, Marco Turchi, José GC de Souza, Daniele Falavigna, and Mohammed RH Qwaider. TranscRater: a Tool for Automatic Speech Recognition Quality Estimation. *ACL 2016*, pages 43–48, 2016.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated Gain-based Evaluation of IR Techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

Frederick Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, MA, USA, 1997. ISBN 0-262-10066-5.

Liangxiao Jiang. Learning random forests for ranking. *Frontiers of Computer Science in China*, 5(1):79–86, 2011. ISSN 1673-7466. doi: 10.1007/s11704-010-0388-5.

Penny Karanasou, Mark J. F. Gales, and Philip C. Woodland. I-vector Estimation using Informative Priors for Adaptation of Deep Neural Networks. In *Proc. of the 9th International Conference on Spoken Language Processing (INTERSPEECH)*, pages 2872–2876, Dresden, Germany, 2015.

Slava Katz. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE transactions on acoustics, speech, and signal processing*, 35(3):400–401, 1987.

Patrick Kenny, Pierre Ouellet, Najim Dehak, Vishwa Gupta, and Pierre Dumouchel. A Study of Interspeaker Variability in Speaker Verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(5): 980–988, 2008.

Christopher Kermorvant. A comparison of noise reduction techniques for robust speech recognition. Idiap-RR Idiap-RR-10-1999, IDIAP, 0 1999. IDIAP-RR 99-10.

Kenichi Kumatani, John McDonough, Jill Fain Lehman, and Bhiksha Raj. Channel Selection Based on Multichannel Cross-correlation Coefficients for Distant Speech Recognition. In *Proc. of the 3rd Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, pages 1–6, Edinburgh, UK, 2011.

Kenichi Kumatani, Takeshi Arakawa, Koji Yamamoto, John McDonough, Bhiksha Raj, Rajdeep Singh, and Ivan Tashev. Microphone Array Processing for Distant Speech Recognition: Towards Real-World Deployment. In *Proc. of Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1–10, Hollywood, CA, USA, 2012.

Lecorvè, Gwènolè, and Petr Motlicek. Conversion of Recurrent Neural Network Language Models to Weighted Finite State Transducers for Automatic Speech Recognition. Technical report, IDIAP, 2012.

Bo Li and Khe C. Sim. Comparison of Discriminative Input and Output Transformation for Speaker Adaptation in the Hybrid NN/HMM Systems. In *Proc. of the 9th International Conference on Spoken Language Processing (INTERSPEECH)*, pages 526–529, Makuhari, Japan, 2010.

Xiang Li, Rita Singh, and Richard M. Stern. Lattice Combination for Improved Speech Recognition. In *Proc. of the International Conference of Spoken Language Processing*, Denver, CO, USA, 2002.

Xiao Li and J. Bilmes. Regularized Adaptation of Discriminative Classifiers. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP)*, Toulouse, France, 2006.

Yoseph Linde, Andres Buzo, and Robert Gray. An Algorithm for Vector Quantizer Design. *IEEE Transactions on communications*, 28(1):84–95, 1980.

Xunying Liu, Yongqiang Wang, Xie Chen, MJF Gales, and PC Woodland. Efficient lattice rescoring using recurrent neural network language models. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4908–4912. IEEE, 2014.

Alessia Mammone, Marco Turchi, and Nello Cristianini. Support vector machines. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1 (3):283–289, 2009. ISSN 1939-0068. doi: 10.1002/wics.49.

Lidia Mangu, Eric Brill, and Andreas Stolcke. Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks. *Computer Speech & Language*, 14(4):373–400, 2000. ISSN 0885-2308.

Brian McFee and Gert R Lanckriet. Metric Learning to Rank. In *Proc. of the 27th International Conference on Machine Learning (ICML)*, pages 775–782, Haifa, Israel, 2010.

Frank McSherry and Marc Najork. Computing Information Retrieval Performance Measures Efficiently in the Presence of Tied Scores. In *Advances in Information Retrieval*, volume 4956, pages 414–421. Springer, 2008. ISBN 978-3-540-78645-0.

X. Mestre and M.A. Lagunas. On Diagonal Loading for Minimum Variance Beamformers. In *Proc. of the 3rd IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 459–462, Darmstadt, Germany, 2003.

Yajie Miao, Hao Zhang, and Florian Metze. Speaker Adaptive Training of Deep Neural Network Acoustic Models using I-vectors. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(11):1938–1949, 2015.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent Neural Network Based Language Model. In *Proc. of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1045–1048, Makuhari, Chiba, Japan, 2010.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. There's No Comparison: Reference-less Evaluation Metrics in Grammatical Error Correction. *arXiv preprint arXiv:1610.02124*, 2016.

Matteo Negri, Marco Turchi, Jos'e G. C. de Souza, and Falavigna Daniele. Quality Estimation for Automatic Speech Recognition. In *Proc. of the 25th International Conference on Computational Linguistics: Technical Papers (COLING)*, pages 1813–1823, Dublin, Ireland, 2014.

Joao Neto, Luis Almeida, Mike Hochberg, Ciro Martins, Luis Nunes, Steve Renals, and Tony Robinson. Speaker Adaptation for Hybrid HMM-ANN Continuous Speech Recognition System. In *Proc. of the 9th International Conference on Spoken Language Processing (INTERSPEECH)*, Madrid, Spain, 1995.

Raymond W. M. Ng, Kashif Shah, Walter Aziz, Lucia Specia, and Hain Hain. Quality Estimation for ASR K-best List Rescoring in Spoken Language Translation. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5226–5230, April 2015a. doi: 10.1109/ICASSP.2015.7178968.

Raymond W. M. Ng, Kashif Shah, Lucia Specia, and Thomas Hain. A Study on the Stability and Effectiveness of Features in Quality Estimation for Spoken Language Translation. In *Conference of the International Speech Communication Association*, INTERSPEECH, Dresden, Germany, 2015b. URL `http://staffwww.dcs.shef.ac.uk/people/W.Ng/Ng_Interspeech2015.pdf`.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Sree H. K. Parthasarathi, Bjorn Hoffmeister, Spyros Matsoukas, Arindam Mandal, Nikko Strom, and Sri Garimella. fMLLR Based Feature-space Speaker Adaptation of DNN Acoustic Models. In *Proc. of the 9th International Conference on Spoken Language Processing (INTERSPEECH)*, pages 3630–3634, Dresden, Germany, 2015.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine Learning in Python. *Machine Learning Research*, 12:2825–2830, 2011.

Thomas Pellegrini and Isabel Trancoso. Improving ASR Error Detection with Non-decoder Based Features. In *Proc. of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1950–1953, Makuhari, Chiba, Japan, 2010.

Michael Pitz, Frank Wessel, and Hermann Ney. Improved Mllr Speaker Adaptation Using Confidence Measures For Conversational Speech Recognition. In *Proc. Int. Conf Spoken Language Processing*, Beijing,China, 2000.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The Kaldi Speech Recognition Toolkit. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Hawaii, USA, 2011.

Lawrence Rabiner and B Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

Lawrence R. Rabiner and Biing H. Juang. *Fundamentals of Speech Recog-*

nition. Prentice-Hall Signal Processing Series: Advanced monographs. PTR Prentice Hall, 1993. ISBN 9780130151575.

Giuseppe Riccardi and Dilek Hakkani-Tur. Active Learning: Theory and Applications to Automatic Speech Recognition. *IEEE Transaction on Speech and Audio Processing*, 13(4):504–511, 2005.

Nicholas Ruiz and Marcello Federico. Phonetically-oriented word error alignment for speech recognition error analysis in speech translation. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 296–302, Dec 2015.

Holger Schwenk. CSLM-a modular open-source continuous space language modeling toolkit. In *Proc. of the 9th International Conference on Spoken Language Processing (INTERSPEECH)*, 2013.

Holger Schwenk and Jean-Luc Gauvain. Improved ROVER using Language Model Information. In *Proc. of ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*, pages 47–52, Orsay, France, 2000.

Frank Seide, Gang Li, Chen Xie, and Dong Yu. Feature Engineering in Context-dependent Deep Neural Networks for Conversational Speech Transcription. In *Proc. of IEEE ASRU Workshop*, Hawaii, USA, 2011.

Mathew Stephen Seigel. *Confidence Estimation for Automatic Speech Recognition Hypotheses*. PhD thesis, University of Cambridge, Cambridge, England, 2013.

Sabato M. Siniscalchi, Jinyu Li, and Chin H. Lee. Hermitian Polynomial for Speaker Adaptation of Confectionist Speech Recognition Systems. *IEEE Trans. on Audio Speech and Language Processing*, 21(10):2152–2161, 2013.

Alex J Smola and Bernhard Schölkopf. A Tutorial on Support Vector Regression. *Statistics and computing*, 14(3):199–222, 2004.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of association for machine translation in the Americas*, volume 6, 2006.

Radu Soricut and Abdessamad Echihabi. TrustRank: Inducing Trust in Automatic Translations via Ranking. In *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 612–621, Uppsala, Sweden, 2010. Association for Computational Linguistics.

Lucia Specia, Nicola Cancedda, Marc Dymetman, Marco Turchi, and Nello Cristianini. Estimating the Sentence-Level Quality of Machine Translation Systems. In *Proc. of the 13th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 28–35, Barcelona, Spain, 2009.

Andreas Stolcke. Making the Most From Multiple Microphones in Meeting Recognition. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4992–4995, Prague, Czech Republic, 2011.

Yik-Cheung Tam, Yun Lei, Jing Zheng, and Wen Wang. ASR Error Detection using Recurrent Neural Network Language Model and Complementary ASR. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2312–2316, Florence, Italy, 2014.

S. Thomas, M.L. Seltzer, K. Church, and H. Hermansky. Deep Neural Network Features and Semi-Supervised Training for Low Resource Speech

Recognition. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6704–6708, Vancouver, Canada, 2013.

Nicola Ueffing and Hermann Ney. Word-Level Confidence Estimation for Machine Translation. *Computational Linguistics*, 33(1):9–40, 2007.

Konstantinos Veropoulos, Colin Campbell, Nello Cristianini, et al. Controlling the sensitivity of support vector machines. In *Proceedings of the international joint conference on AI*, pages 55–60, 1999.

Karel Vesely, Arnab Ghoshal, Lukas Burget, and Daniel Povey. Sequence-discriminative Training of Deep Neural Networks. In *Proc. of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2345–2349, Florence, Italy, 2011.

Emmanuel Vincent, Jon Barker, Shigetaka Watanabe, Jonathan Le Roux, Francesco Nesta, and Marco Matassoni. The Second CHiMEspeech Separation and Recognition Challenge: An Overview of Challenge Systems and Outcomes. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 162–167, Olomouc, Czech Republic, 2013.

Ye-Yi Wang, A. Acero, and C. Chelba. Is word error rate a good indicator for spoken language understanding accuracy. In *Automatic Speech Recognition and Understanding, 2003. ASRU '03. 2003 IEEE Workshop on*, pages 577–582, Nov 2003a.

Zhirong Wang, Tanja Schultz, and Alex Waibel. Comparison of acoustic model adaptation techniques on non-native speech. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 1, pages I–540. IEEE, 2003b.

Frank Wessel, Ralf Schluter, Klaus Macherey, and Hermann Ney. Confidence Measures for Large Vocabulary Continuous Speech Recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 9(3):288–298, 2001.

Matthias Wölfel and John McDonough. Combining Multi-Source Far Distance Speech Recognition Strategies. In *Proc. of the 9th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 3149–3152, Lisbon, Portugal, 2005.

Matthias Wölfel and John McDonough. *Distant Speech Recognition*. John Wiley & Sons Ltd, 2009. ISBN 978-0-470-51704-8.

David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

J. Yang, C. Zhang, A. Ragni, M. J. F. Gales, and P. C. Woodland. System combination with log-linear models. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5675–5679, March 2016. doi: 10.1109/ICASSP.2016.7472764.

K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong. Adaptation of Context-Dependent Deep Neural Networks for Automatic Speech Recognition. In *Proc. of SLT*, Miami, Florida, USA, 2012.

Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. The htk book. *Cambridge university engineering department*, 3: 175, 2002.

D. Yu, K. Yao, H. Su, G. Li, and F. Seide. KL-Divergence Regularized Deep Neural Network Adaptation for Improved Large Vocabulary Speech Recognition. In *Proc. of the IEEE International Conference on Acoustics,*

Speech, and Signal Processing (ICASSP), pages 7893–7897, Vancouver, Canada, 2013.

Hamed Zamani, José GC de Souza, Matteo Negri, Marco Turchi, and Daniele Falavigna. Reference-free and Confidence-independent Binary Quality Estimation for Automatic Speech Recognition. In *Proceedings of the Second Italian Conference on Computational Linguistics CLiC-it 2015*, Trento, Italy, December 2015.

Puming Zhan and Alex Waibel. Vocal tract length normalization for large vocabulary continuous speech recognition. Technical report, CMU COMPUTER SCIENCE TECHNICAL REPORTS, 1997.

Rong Zhang and Alexander I. Rudnicky. Investigations of Issues for Using Multiple Acoustic Models to Improve Continuous Speech Recognition. In *Proc. of the 9th International Conference on Spoken Language Processing (INTERSPEECH)*, Pittsburgh, PA, USA, 2006.

Ke Zhou, Gui-Rong Xue, Hongyuan Zha, and Yong Yu. Learning to Rank with Ties. In *Proc. of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–282, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4.