Trigger-Based Language Model Adaptation for Conversational Speech Transcription

by

Carlos TRONCOSO ALARCÓN

submitted to
Japan Advanced Institute of Science and Technology
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Supervisor: Professor Jianwu DANG

School of Information Science
Japan Advanced Institute of Science and Technology

March, 2006

Abstract

When humans are learning a second language, the presence of keywords in foreign speech helps the non-native speaker to recognize the subject matter of the conversation and, consequently, recall the already acquired vocabulary that belongs to the corresponding topic, thus facilitating the comprehension of the foreign language. Computers are like non-native speakers when it comes to automatic speech recognition, because they often misrecognize what they "hear", so we also expect topic information characterized by related keywords to aid the recognition process in this case.

Recently, the major target of automatic speech recognition research has shifted from dictation of document-style sentences to transcription of spontaneous conversational-style speech. Research in this field is still immature, and the current recognition accuracy is low. Language models play a crucial role in automatic speech recognition since they provide effective constraint and preference for possible word sequences. Without language models, speech recognizers would blindly choose among candidate words without any linguistic criterion. The most widely used language model is the n-gram model, which models the occurrence probability of n consecutive words in the text. n-gram models are powerful in modeling short-distance dependencies between words, but cannot capture long-distance dependencies because they rely on a word history limited to $n-1$ words. This thesis addresses the trigger-based language model. This model is a good complement of the n-gram language model, because it incorporates long-distance topic constraints by means of related keywords, called *trigger pairs*. Meetings and conversations, which are the main target of this study, are centered in a topic in many cases, so the trigger pairs could capture long-distance topic constraints in these tasks. The trigger-based language model is also insensitive to disfluencies, which are prominent characteristics in conversational speech, because it focuses on the co-occurrence of topic keywords.

However, reliable statistical estimation is the most critical problem for this kind of long-distance language model, especially for spontaneous speech, where only a small amount of training data is available compared with document-style language. This work proposes two methods to fully exploit the available in-domain data to adapt the trigger-based language model to conversational speech. Here, task-dependent trigger pairs are extracted that match more closely the addressed task. In addition, to enhance the reliability of probability estimates derived from the small amount of data, a back-off scheme that incorporates the statistics from a large corpus is proposed.

Chapter 1 introduces the two main approaches to language modeling and the application of statistical language modeling to automatic speech recognition.

Chapter 2 reviews the major language modeling techniques and presents the concept of the proposed approach. Then, the evaluation measures for language model performance and the different ways of incorporating long-distance language models are explained.

Chapter 3 presents a trigger-based language model for the transcription of travel expressions and extemporaneous speeches on given topics. Generally in language modeling, when the training corpus matches the target task its size is typically small, and

therefore insufficient to provide reliable probability estimates. On the other hand, large corpora are often too general to capture task dependency. The proposed approach tries to overcome this generality-sparseness trade-off problem by constructing a trigger-based language model in which task-dependent trigger pairs are first extracted from the corpus that matches the task, and then their occurrence probabilities are estimated from both the task corpus and a large text corpus to avoid the data sparseness problem. In the experiments, the perplexity by the proposed model was lower than that by the conventional trigger-based model constructed from one single corpus, and 12.8% lower than the baseline.

Chapter 4 addresses the trigger-based language model for the transcription of panel discussions on political and economic issues. Here, the previous approach cannot be used because of the lack of in-domain training data. In meetings, the topic is focused and consistent throughout the whole session, therefore keywords can be correlated over long distances. The trigger-based language model can capture such long-distance dependencies, but the derived trigger pairs are not task-dependent if it is typically constructed from a large general corpus. The proposed method makes use of the initial speech recognition results to extract task-dependent trigger pairs and to estimate their statistics. Moreover, the back-off scheme is introduced to exploit the statistics estimated from a large corpus. The proposed model reduced the perplexity considerably more than the typical trigger-based language model constructed from a large corpus, and achieved a remarkable perplexity reduction of 44% over the baseline when combined with an adapted trigram language model. In addition, a reduction in word error rate was obtained when using the proposed language model to rescore word graphs.

Chapter 5 concludes the thesis with a summary of contributions and future directions.

Acknowledgments

The present research would have been impossible to complete without the help of many people.

First and foremost, I wish to express my most sincere gratitude to my principal advisor, Professor Tatsuya Kawahara of Kyoto University, for inviting me to join his lab in first place, for his priceless guidance and advice during my research, for the uncountable things I learned from him, and also for his help during my job search.

I would like to thank my advisor Professor Jianwu Dang of Japan Advanced Institute of Science and Technology for his helpful suggestions and support.

I also wish to express my gratefulness to Dr. Hirofumi Yamamoto and Dr. Genichiro Kikui of Advanced Telecommunications Research Institute International (ATR) for their fruitful guidance during the beginning of this research during my internship at ATR, and for their valuable comments.

I am very grateful to Dr. Shinsuke Mori of IBM Tokyo Research Laboratory for his helpful suggestions and discussion.

I devote my sincere thanks to Dr. Yuya Akita of Kyoto University for his constant help and support throughout this work, and for providing me with many helpful tools for my experiments.

I would like to thank Dr. Nobuhiro Kaji and Professor Sadao Kurohashi of the University of Tokyo for providing me with the conversational web corpus.

I would also like to give thanks to Professor Hiroshi Shimodaira of the University of Edinburgh for his guidance during the Master's course and the start of the doctoral course.

I am grateful to Professor Kentaro Torisawa, Professor Kiyoaki Shirai, and Professor Isao Tokuda for revising my work and for their valuable comments.

I thank researchers at IBM TJ Watson Research Center, Microsoft, and IBM Tokyo Research Laboratory for their fruitful comments on my work during my presentations in their respective locations.

I owe a considerable debt of gratitude to my wife, Remedios García Bonilla, for her encouragement, support, and patience during my research.

I am particularly thankful to God, who helped me in every moment.

Last, but not least, I would also like to take this opportunity to thank all my fellows in Professor Kawahara's laboratory for their sincere help and cooperation, and my family and friends for their constant support and encouragement.

To Reme

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

When humans are learning a second language, the presence of keywords in foreign speech helps the non-native speaker to recognize the subject matter of the conversation and, consequently, retrieve the already acquired vocabulary that belongs to the corresponding topic, thus facilitating the comprehension of the foreign language. For example, when Japanese students of English hear the word "pitcher", they will immediately recognize the topic "baseball", and they will expect words like "catcher" or "base" to come up afterwards. Figure 1.1 shows another example of conversation in which the topic facilitates the comprehension of the foreign speech.

Computers can be thought of as non-native speakers when it comes to automatic speech recognition (ASR), because they often misrecognize what they "hear", that is the input speech, so we can expect topic information characterized by related keywords to aid the recognition process in this case too.

Language models are an important and necessary part of ASR systems, because they model the linguistic relations among words in the utterance that is to be recognized. Without language models, speech recognizers would blindly choose among candidate words without any linguistic criterion, resulting in ungrammatical and nonsensical sentences in most cases. The most widely used language model in ASR is the n-gram model. n-grams model the occurrence probability of n consecutive words in the text, and their parameters are estimated from a large text corpus. n-gram models are powerful in modeling short-distance dependencies between words, but they cannot capture long-distance dependencies such as topic information, because they rely on a word history limited to $n-1$ words, where n typically ranges from 2 to 4. Nevertheless, it has proved very difficult to outperform these models, mainly due to their simplicity.

There are some alternative language models that try to overcome this limitation of n-grams. Examples of those that make use of long-distance topic information are the trigger-based language model, the cache-based language model, and latent semantic analysis-based language models. This thesis focuses on the trigger-based language model, which is capable of capturing long-distance dependencies between words. The trigger-based language model uses a set of correlated word pairs, known as *trigger pairs*, to raise the probability of the words "triggered" by others in the word history. The trigger-based language model has been mainly applied to the recognition of newspaper tasks, and it has been typically constructed from large corpora such as newspaper articles. This kind of

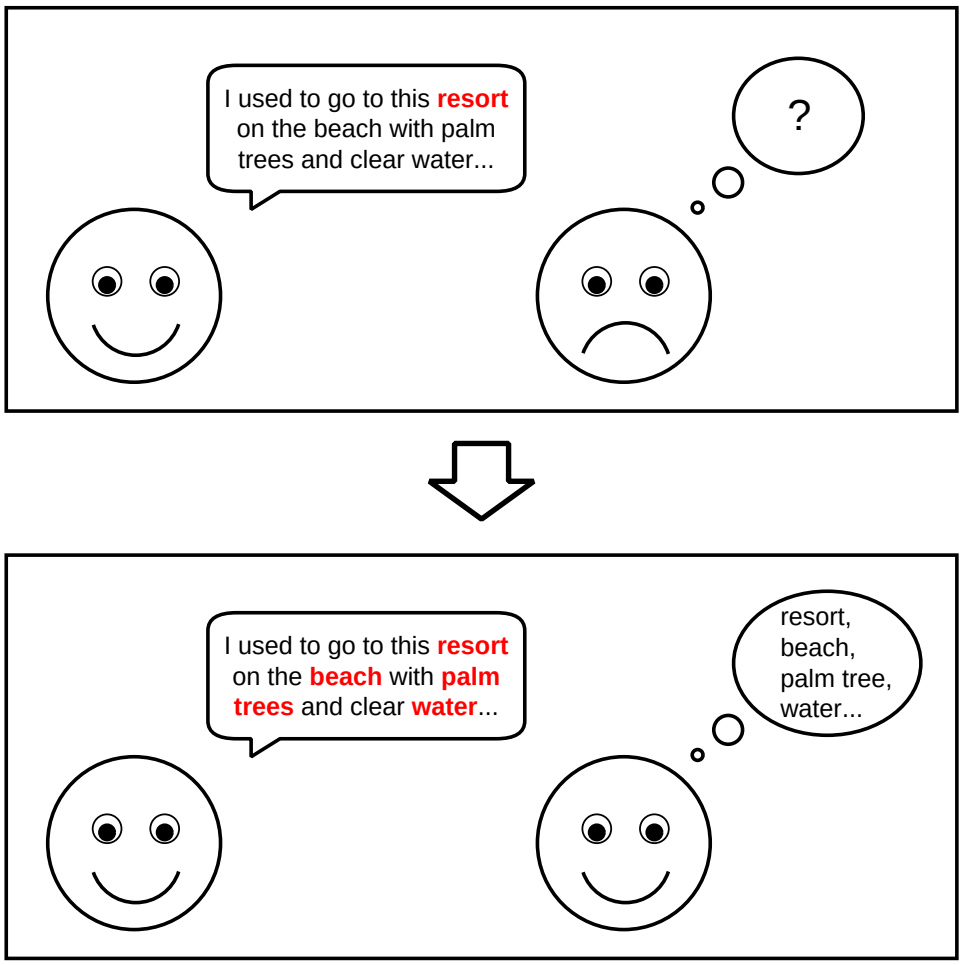

Figure 1.1: Example of conversation in which the topic *resort* facilitates the comprehension of the words *beach, palm, trees*, and *water*.

corpora is usually too general in topic and does not closely match the specific test data, thus the trigger pairs constructed from them are not task dependent.

Language model adaptation tries to improve language modeling by creating language models close in style or topic to the target task. In this research, the trigger-based language model is used to adapt a baseline language model to the target domain by exploiting the available in-domain data to try to take advantage of topic information during the speech recognition process.

The following sections 1.2 and 1.3 introduce language modeling in general as well as its application to ASR. Then, section 1.4 deals with the problems addressed by this research, and section 1.5 describes the organization of this thesis.

1.2 Language modeling

Language modeling is the attempt to characterize, capture and exploit regularities in natural language [55]. Natural language is extremely difficult to model formally, due to its inherent variability and uncertainty.

There are two main approaches to language modeling: statistical language modeling and knowledge-based language modeling. The statistical approach tries to capture regularities in language from large amounts of text in a process known as *training*. On the other hand, knowledge-based modeling uses a set of linguistic rules coded by experts, as well as domain knowledge, to assess the grammaticality of sentences.

The advantages of statistical language modeling over the knowledge-based approach are:

- Statistical models assign a probability to each possible sentence, while knowledge-based models usually only provide a "yes"/"no" answer to the grammaticality of a sentence. Probabilities convey much more information than such a simple answer. Moreover, spoken language is often ungrammatical.
- Statistical models can be inexpensively built from a great variety of domains, as soon as the training procedure has been implemented.
- Coding linguistic rules by hand can be tedious, often incomplete, and sometimes erroneous.
- At runtime, knowledge-based models like parsers are more computationally expensive than statistical models.

Statistical language modeling has also some disadvantages:

- They do not capture the meaning of the text. Therefore, they may assign a high probability to nonsensical sentences. Nevertheless, this kind of sentences can be sometimes found in spontaneous speech due to disfluencies or sudden termination.
- Statistical models require large amounts of training data, which are not always available. However, these language models can also take advantage of smaller training sets through language model adaptation.
- Statistical language modeling often do not make use of linguistic and domain knowledge, which sometimes can be very helpful.

Language modeling is useful, and often crucial, in areas like ASR, machine translation (MT), spelling correction, handwriting recognition, optical character recognition (OCR), document classification, information retrieval, and any other application that process natural language with incomplete knowledge.

In this work, statistical language modeling is used for ASR, because it is less expensive than knowledge-based language modeling and better suited for spoken language tasks, which is the target of this thesis.

1.3 Language models in automatic speech recognition

ASR deals with the problem of automatically transcribing speech into text. ASR is typically performed as follows. First, a preprocessor generates a set of feature vectors

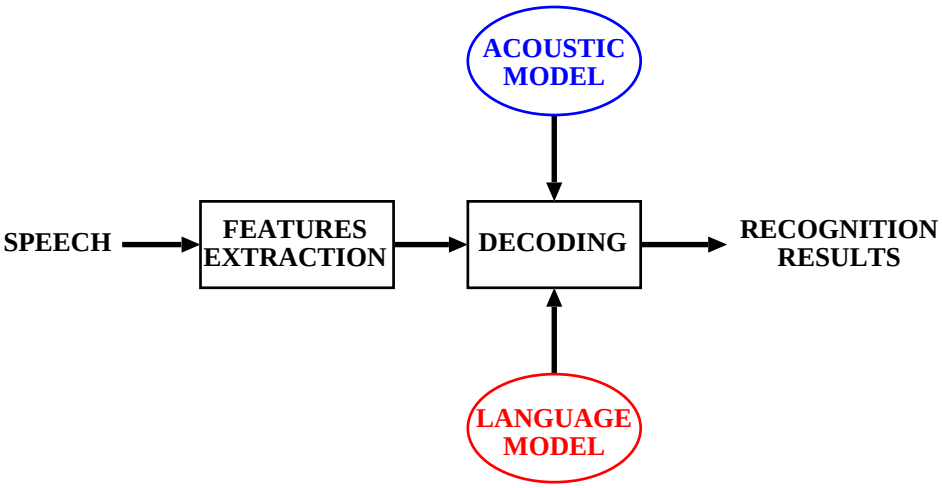

Figure 1.2: The automatic speech recognition paradigm.

which capture the spectral characteristics of the input speech signal (acoustic waveform) at discrete time intervals. Then, these feature vectors are passed to the decoder, which, based on the acoustic and the language model probabilities, searches for the string of words that best matches these vectors. The result of this search is a list of hypothesized transcriptions, which is the output of the ASR system. This paradigm is illustrated in figure 1.2.

The most successful approach to ASR is the statistical one proposed in [2]. The aim is to find the word sequence $\hat{W}$ that maximizes the probability of a word sequence W given the observed acoustic signal A. Applying the Bayes rule:

$$\hat{W} = \arg\max_{W} P(W|A) = \arg\max_{W} \frac{P(A|W)P(W)}{P(A)} = \arg\max_{W} P(A|W)P(W) \tag{1.1}$$

The calculation of $P(A|W)$ is the role of the acoustic model, whereas the language model is responsible for the computation of $P(W)$.

Let $W = w_1^n \triangleq w_1, w_2, ..., w_n$, where the w_i's are the words that make up the word sequence. $P(W)$ can be decomposed, by using the chain rule, in the following way:

$$P(W) = \prod_{i=1}^{n} P(w_i|w_1^{i-1}) \tag{1.2}$$

Most statistical language models try to estimate expressions of the form $P(w_i|H)$, where $H = w_1^{i-1}$ is known as the *history*.

Since the number of possible histories that can precede a given word is very large, it is unfeasible to try to estimate the probability of all of them from the limited corpora that are available. Therefore, some simplification must be applied to the above equation. Usually, the event space is partitioned in equivalence classes depending on some property of the history, that is, we use a function $\phi(H)$. As a result, the simplified equation

becomes:

$$P(W) \approx \prod_{i=1}^{n} P(w_i|\phi(h)) \tag{1.3}$$

For instance, in the *trigram* (3-gram) model the partition ϕ is based on the last two words of the history.

1.4 Problems addressed by this thesis

Recently, the major target of automatic speech recognition research has shifted from dictation of document-style sentences to transcription of spontaneous conversational-style speech. Research in this field is still immature, and the current recognition accuracy rates are low. Therefore, more effort should be devoted to devise new technologies that contribute to further progress in this field.

Meetings and conversations, which are the main target of this study, are centered in a topic in many cases, so the trigger-based language model could be used to capture long-distance topic constraints in these tasks. The trigger-based language model is also insensitive to disfluencies, because it focuses on the co-occurrence of topic keywords. Disfluencies (filled pauses, repetitions, repairs...) are a kind of phenomenon often found in spontaneous speech that disrupts the smooth flow of the discourse. They are a serious problem for language modeling, because they can make sentences ungrammatical, contribute to data sparseness, and make dependencies between words longer.

Although the trigger-based language model seems appropriate for conversational speech, its reliable statistical estimation is the most critical problem, especially for this kind of corpora. Conversational text corpora are expensive to produce, as compared to written-style text corpora, so the available amount of training data is usually insufficient to derive reliable task-dependent language models.

This work proposes two methods to fully exploit the available in-domain data to adapt the trigger-based language model to conversational speech. In both methods, task-dependent trigger pairs that match more closely the addressed task are extracted from the in-domain data. In the first approach, the available training data is used to extract the trigger pairs, while in the second approach the initial speech recognition results are used for this purpose. In addition, to enhance the reliability of probability estimates derived from the small amount of available data, a back-off scheme that incorporates the statistics from a large corpus to the model is proposed.

1.5 Thesis organization

The rest of this thesis is organized as follows. First, chapter 2 presents a review of conventional statistical language modeling techniques. Then, the concept of the proposed approach is introduced. This is followed by the explanation of the main measures for language model evaluation and common methods for the incorporation of language models in the ASR system. Chapter 3 describes the application of the trigger-based language model to two different conversational speech tasks by obtaining the trigger pairs from the target corpus and estimating their probabilities from both this task corpus and a large

corpus, and then combining these probabilities by means of a back-off model. Chapter 4 proposes a different adaptation scheme based on the extraction of trigger pairs from the initial speech recognition results and also a back-off model using the probabilities estimated from the recognition results and a large corpus. Finally, chapter 5 concludes this thesis by summarizing and giving future research directions for it.

Chapter 2

Overview

2.1 Introduction

This chapter presents an overview of both the basic language modeling theory and the proposed approach. First, section 2.2 explains the major language modeling techniques. Then, the proposed approach is presented in section 2.3. The two most important evaluation measures for language model performance are introduced in section 2.4, followed by the different integration methods of language models in the ASR system in section 2.5.

2.2 Review of conventional language models

Many different language models have been proposed in the literature. Below is a description of the most interesting approaches classified by the length of the scope they cover.

2.2.1 Short distance

Word n-grams

A word n-gram [2] is a model that uses the last $n-1$ words of the history as its sole information source. Typically n equals 2 to 4, and they are called *bigram*, *trigram*, and *4-gram* models, respectively.

As commented in the previous chapter, n-gram models partition the data into equivalence classes based on the last $n-1$ words of the history. Therefore, the following simplification is made:

$$P(w_i|w_1^{i-1}) \approx P(w_i|w_{i-n+1}^{i-1}) \tag{2.1}$$

In this way, a bigram estimates $P(w_i|H)$ by $P(w_i|w_{i-1})$, a trigram by $P(w_i|w_{i-2}, w_{i-1})$, and so on.

The probabilities of an n-gram model are estimated from large amounts of text data by the relative frequency of appearance of the tuple, that is:

$$f(w_i|w_{i-n+1}^{i-1}) = \frac{N(w_{i-n+1}, \cdots w_{i-1}, w_i)}{N(w_{i-n+1}, \cdots w_{i-1})} \tag{2.2}$$

where $N(W)$ denotes the number of times the tuple W is observed in the training data.

n-grams are affected by the classic modeling trade-off between detail and reliability. When n is small, the parameters are reliably estimated from the training data, because the tuples are found easily. However, the modeling power is smaller than for greater values of n. On the other hand, when n is big, the data are insufficient and the estimates become unreliable.

Some smoothing techniques such as *deleted interpolation* [29] or *back-off* [35, 42] have been proposed to assign proper probabilities to events that were not seen during training.

Deleted interpolation consists of linearly interpolating an n-gram model with lower-order n-grams down to the unigram. For example, a trigram probability $P(w_i|w_{i-2}, w_{i-1})$ may be estimated as:

$$\begin{aligned} P(w_i|w_{i-2}, w_{i-1}) = \lambda_3(w_{i-2}, w_{i-1}) f(w_i|w_{i-2}, w_{i-1}) + \lambda_2(w_{i-2}, w_{i-1}) f(w_i|w_{i-1}) \\ + \lambda_1(w_{i-2}, w_{i-1}) f(w_i) + \lambda_0 \end{aligned} \tag{2.3}$$

where the history-dependent weights λ_j are chosen to maximize the likelihood of some held-out data, and satisfy:

$$\sum_{j=0}^{3} \lambda_j = 1 \tag{2.4}$$

for each history.

Back-off smoothing uses lower-order n-grams with enough evidence to approximate higher-order n-grams with insufficient evidence. For example, a trigram model is estimated as:

$$P(w_i|w_{i-2}, w_{i-1}) = \begin{cases} f(w_i|w_{i-2}, w_{i-1}), & \text{if } N(w_{i-2}, w_{i-1}, w_i) > T \\ Q_T(w_i|w_{i-2}, w_{i-1}), & \text{if } 1 \leq N(w_{i-2}, w_{i-1}, w_i) \leq T \\ \alpha(w_{i-2}, w_{i-1}) P(w_i|w_{i-1}), & \text{otherwise} \end{cases} \tag{2.5}$$

where Q_T is a discounting function, T is a threshold, and α is the remaining probability mass for all the unseen w_i.

The choice of n in n-grams should depend on the amount of data available. For the sizes of the corpora typically available nowadays, trigrams own the best balance between reliability and detail, although interest is gradually moving towards 4-grams and beyond.

n-gram models are easy to implement and easy to interface to the ASR decoder. They are very powerful and difficult to improve, mainly because of their simplicity. They seem to capture well short-range dependencies. It is for these reasons that they have become the standard language models in ASR.

Unfortunately, they also have their drawbacks. First, they are unaware of any phenomenon or constraint that is outside their limited scope. Therefore, they may assign high probabilities to nonsensical and even ungrammatical utterances, as long as they satisfy local constraints. In addition, the predictors in n-gram models are defined by their order in the sentence, not by their linguistic properties. Therefore, histories like "the fireman extinguished the" and "the fireman extinguished quickly the" are very different for a trigram, even though they are very likely to precede the same word.

Class-based n-grams

Class-based n-grams [7] are n-grams whose parameter space has been reduced by clustering the words into classes. The n-grams are then based on these classes, rather than the words themselves.

If it is assumed that each word w belongs to only one class $g(w)$, then this model can take many forms, for example,

$$P(w_i|H) = P(w_i|g(w_{i-2}), g(w_{i-1})) \tag{2.6}$$
$$P(w_i|H) = P(w_i|g(w_{i-2}), w_{i-1}) \tag{2.7}$$
$$P(w_i|H) = P(g(w_i)|g(w_{i-2}), g(w_{i-1}))P(w_i|g(w_i)) \tag{2.8}$$

In practice, it is the last one that is the most used in class-based n-grams.

The clustering method itself can also take many forms. Firstly, the clustering can be based on the linguistic knowledge. The best known example of this method is clustering by part of speech (POS). POS clustering attempts to capture syntactic dependencies between adjacent words in the text. This approach has several problems, though: some words can belong to more than one POS, POS classifications made by linguists may not be optimal for language modeling, and there are many different schemes for POS classification.

In second place, in clustering by domain knowledge, all words that will behave in a similar fashion are manually grouped together. For example, days of the week, numbers, etc. This approach can be especially helpful when the amount of training data is limited.

Finally, in data-driven clustering, a large amount of data is used to automatically derive classes by statistical means. This is often better than clustering by hand based on one's intuition. However, reliance on data instead of on external knowledge sources can also be problematic. For example, if the amount of training data available is not large enough, the resulting classes may not be reliable. The ideal data-driven clustering would be one supervised by an expert.

Class-based n-grams have advantages over the basic n-grams. Since the possible number of histories is reduced, the model becomes more compact. Therefore, it could be expanded to include more context. For example, a class-based 4-gram model might be approximately the same size as a trigram. In addition, since the number of classes is generally smaller than the size of the vocabulary, the data sparseness is reduced, and even if a word n-gram is not found in the training data, the equivalent class-based n-gram is likely to have been seen. For this reason, these models have been very helpful in situations where the training data available were limited.

The disadvantage of these models is that they lose some of the semantic information that word n-grams capture. For example, the word trigram "Sunday school teacher" captures the semantic relations between *Sunday, school*, and *teacher*, which cannot be captured by class trigrams. This can be partially overcome by constructing language models that incorporate information from both word and class-based n-grams. A more important drawback of class-based n-grams is that they do not solve the locality problem of n-grams.

Mixture-based language models

These models [11, 27] are composed of several language models, each of which is specific to a particular topic or sub-language. The probability distributions from these component

language models are linearly interpolated to form the global language model probability. The interpolation weights reflect, at each moment, which component sub-language is currently being emphasized.

Let $M_1, M_2, ..., M_k$ be the component language models. The overall language model probability is then

$$P(w_i|H) = \sum_{j=1}^{k} \lambda_j P_{M_j}(w_i|H) \tag{2.9}$$

where the λ_j's are the interpolation weights, with values such that

$$\sum_{j=1}^{k} \lambda_j = 1 \tag{2.10}$$

Usually, the first step when creating a mixture-based language model is the clustering: the training data has to be partitioned in homogeneous components. This can be done automatically, with some iterative clustering algorithm, or manually, according to the topic, style of text, etc.

The number of clusters in which the training data should be partitioned is a delicate matter. A number too small will result in a model incapable of discerning between topics or linguistic styles in detail. Too large a number will lead to a bunch of undertrained models with poor probability estimates. It is common that one of the components be the whole training data, in order to smooth the estimates and avoid data fragmentation.

The next step is typically to construct an n-gram model for each of the constituents. Then, the interpolation weights can be calculated by using the expectation maximization (EM) algorithm [18] in such a way as to maximize the likelihood of some held-out data.

These language models are theoretically very attractive and represent a sound approach to language model adaptation. However, they still have the short-scope limitation of n-grams, and they have not significantly improved speech recognition accuracy so far.

2.2.2 Intermediate distance

Long-distance n-grams

These models [25] attempt to capture the dependencies between the predicted word and $n-1$-grams that are some distance back in the history. For instance, a distance-2 trigram predicts w_i based on (w_{i-3}, w_{i-2}). Distance-1 n-grams are consequently the conventional n-grams themselves.

These models have very serious limitations. Even though they capture dependencies between words that are separated by distance d, they cannot merge training instances that use different values of d, therefore, they unnecessarily fragment the training data. In other words, they do not pay attention to the nature of the text in order to decide an appropriate value for d, but they simply skip the words that are nearer than d words back in the history.

2.2.3 Long distance

Cache-based language model

This model [44, 45] is based on the observation that a word that has appeared recently in the history has a high probability of reappearing.

A cache memory similar to that of computers is used to store the words of recent appearance. The word probabilities are estimated from their recent frequency of use. If a candidate word is in the cache, its probability is raised.

Typically, a cache-based component is linearly interpolated with an n-gram language model:

$$P(w_i|H) = \lambda P_{cache}(w_i|H) + (1 - \lambda)P_{n-gram}(w_i|H) \tag{2.11}$$

Usually, a cache of the last K words is maintained, and the cache-based probability of a word is computed as the unigram probability of the word within the cache, that is,

$$P_{cache}(w_i|H) = \frac{N_{cache}(w_i)}{K} \tag{2.12}$$

where $N_{cache}(w)$ is the number of times w appears in the cache.

The original cache-based model was interpolated with a class-based trigram based on the POS, and a cache of size 200 was maintained for each POS. The interpolation weights were calculated individually for each POS.

Several extensions have been proposed to this language model, being the most obvious the addition of the cache-based component to a word-based trigram, rather than a class-based model [27].

The cache need not be limited to containing single words. Instead, recent bigrams and trigrams can also be incorporated to the cache and their probabilities boosted [31]. This approach has the problem that the probabilities of n-grams in the cache cannot be reliably estimated due to the insufficient information contained in several hundred words back.

Another extension used the idea that the more recent words are more influential in predicting forthcoming words than those in the more distant past [11]. With this in mind, an exponentially decaying cache was constructed. This is a cache in which the probability of the words inside the cache decay exponentially with the distance from the word being predicted.

The cache-based language model significantly reduces the perplexity of standard language models, and some of the extensions mentioned above contributed to a further improvement in terms of perplexity. However, the same does not apply to recognition accuracy, which has not been noticeably improved by this model so far. This is because during the speech recognition experiments the word history is erroneous, so the cache-based model helps to propagate the errors to the succeeding utterances.

Trigger-based language model

The trigger-based language model [47, 55, 56], like the cache-based model, also uses a cache memory of recent words. However, contrary to the original cache-based model, only "rare" words are incorporated to the cache. A word is defined as rare relative to a threshold of static unigram frequency.

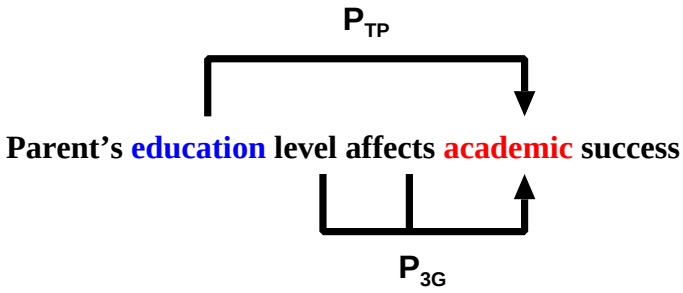

Figure 2.1: Example of long-distance dependency captured by the trigger-based language model but not by the trigram model.

In order to extract information from the document history, a basic information bearing element called *trigger pair* is used. If a word a is semantically well correlated with another word b, then $a \rightarrow b$ is called a trigger pair, with a being the triggering word and b the triggered word. The occurrence of a in the word history triggers the appearance of b, that is, if a appears in the text, the model will predict a heightened probability for b.

Figure 2.1 illustrates an example of long-distance dependency captured by the trigger-based language model but not by the widely used trigram model. In the example, the trigger pair *education* $\rightarrow$ *academic* is used to help predict *academic*, a dependency that falls out of the scope of the trigram model in this case.

The trigger pairs are created from a large text corpus by using the average mutual information measure between the triggering word a and the triggered word b:

$$\begin{aligned} I(a;b) = P(a,b)\log\frac{P(b|a)}{P(b)} + P(a,\bar{b})\log\frac{P(\bar{b}|a)}{P(\bar{b})} \\ + P(\bar{a},b)\log\frac{P(b|\bar{a})}{P(b)} + P(\bar{a},\bar{b})\log\frac{P(\bar{b}|\bar{a})}{P(\bar{b})} \end{aligned} \tag{2.13}$$

Here, $\bar{a}$ means "any word different from a". A high average mutual information indicates that the appearance of b is highly correlated with the appearance of a.

The model is usually formulated as a constraint of a maximum entropy (ME) framework [17, 28] in which n-grams, long-distance n-grams and so on can also take part as constraints of the model, although there are works in which linear interpolation is used to combine the baseline n-gram model with the trigger-based model [73, 74, 80, 5]. In this thesis we adopt the latter approach, because it is simpler and because ME suffers from very long training times.

Latent semantic analysis-based language model

Latent semantic analysis (LSA) [16] is an algebraic technique that can be used to infer the latent semantic relationship among words by means of their co-occurrence in identical contexts. Given a text corpus $\mathcal{T}$ of N documents, with a vocabulary $\mathcal{V}$ of M words, LSA defines a mapping between the discrete sets $\mathcal{T}$ and $\mathcal{V}$ and a continuous vector space $\mathcal{S}$. A document here is a semantically cohesive set of words such as a sentence, paragraph, newspaper article, etc.

The first step is to construct a word-document co-occurrence matrix W, with rows corresponding to words in $\mathcal{V}$ and columns to documents in $\mathcal{T}$, where word order is ignored. Each element in W is the word count in the corresponding document normalized for document length and word entropy:

$$w_{ij} = (1 - \epsilon_i)\frac{c_{i,j}}{n_j} \tag{2.14}$$

where $c_{i,j}$ is the number of times word w_i occurs in document d_j, n_j is the total number of words present in d_j, and ϵ_i is the normalized entropy of w_i in $\mathcal{T}$, given by:

$$\epsilon_i = -\frac{1}{\log N}\sum_{j=1}^{N}\frac{c_{i,j}}{t_i}\log\frac{c_{i,j}}{t_i} \tag{2.15}$$

where $t_i = \sum_j c_{i,j}$ is the total number of times w_i occurs in $\mathcal{T}$.

The second step is to reduce the dimensionality of the resulting large sparse matrix by applying order-R singular value decomposition (SVD):

$$W \approx \hat{W} = USV^T \tag{2.16}$$

where $U_{M\times R}$ is a left singular word matrix, $S_{R\times R}$ is a diagonal matrix of singular values, and $V_{N\times R}$ is a right singular document matrix. This transformation captures the major structural associations in W and removes noise. Values of R in the range of 100 to 300 are typically used for information retrieval. The R-dimensional representations of the word and document vectors are given by $u_i S$ and $v_j S$, respectively, where u_i and v_j are the corresponding rows of U and V. Any new document d can be considered as an additional column of the matrix W, and can be represented as $v = d^T U$.

In the LSA-based language model [3], the closeness between a word w_q and its associated history, that is the current document so far, represented as d_{q-1}, is measured by the cosine of the angle between $u_q S^{1/2}$ and $v_{q-1}S^{1/2}$:

$$K(w_q, d_{q-1}) = \cos(u_q S^(1/2), v_{q-1}S^{1/2}) \tag{2.17}$$

Since the range of this distance measure is within [-1,1], we need to transform it to a probability measure. One way to do this is as follows:

$$P_{LSA}(w_q|d_{q-1}) = \frac{\pi - \cos^{-1}(K(w_q|d_{q-1}))}{\sum_{w_k}[\pi - \cos^{-1}(K(w_k|d_{q-1}))]} \tag{2.18}$$

The LSA-based language model is usually combined with an n-gram model. The combination proposed in [3] is the following:

$$P(w_q|H) = \frac{P_{n-gram}(w_q|w_{q-n+1}^{q-1})\frac{P_{LSA}(w_q|d_{q-1})}{P(w_q)}}{\sum_{w_i\in\mathcal{V}}[P_{n-gram}(w_i|w_{q-n+1}^{q-1})\frac{P_{LSA}(w_i|d_{q-1})}{P(w_i)}]} \tag{2.19}$$

The LSA-based language model effectively captures large-span semantic relations among words and has proved successful in terms of perplexity and word error rate reduction. However, strictly speaking it is not a probabilistic model, because it requires heuristics to compute the probability of an unseen document. Another approach known as probabilistic latent semantic analysis (PLSA) has been applied to language modeling to account for this [22]. Here, documents are represented as sets of word occurrence probabilities. The problem with the PLSA-based language model is that it can suffer from the overfitting problem [72].

2.3 Proposed approach

2.3.1 Trigger-based language model

This thesis focuses on the trigger-based language model, which is a good complement of the standard n-gram language model because it effectively exploits long-distance dependencies by means of related keywords (trigger pairs). Research on trigger-based language models is insufficient, and they are simpler and easier to implement than other more complex topic-dependent approaches such as LSA-based language models.

The drawback of trigger pairs is that far more information is contained in self-triggers (words that trigger themselves) than in any others; even non-self-triggers tend to be triggers with the same stem (e.g. abuse, abused, abusing). Self-triggers are virtually equivalent to the cache-based language model, so the original trigger-based language model does not significantly outperform the cache-based model. In addition, trigger pairs are usually constructed from a text window of fixed length with the average mutual information measure. This window limits the scope of the dependencies that the trigger-based language model can capture. Therefore, the model captures local topic constraints, rather than global.

In this research, instead of the average mutual information measure, we use the term frequency/inverse document frequency (TF/IDF) to extract the trigger pairs from the whole document, rather than a text window, to capture topic constraints global to the document.

2.3.2 Transcription of conversational speech

This thesis deals with the automatic transcription of conversational speech. Recently, in the speech recognition community, the interest has shifted from written language-style tasks to the recognition of spontaneous speech, which is a field that poses many more challenging problems. Research in this field is still immature, and the current word recognition accuracy rates are low, as opposed to dictation systems or written-style tasks. Therefore, more effort should be devoted to devise new technologies that contribute to further progress in this field.

So far, the trigger-based language model has been mainly applied to the recognition of newspaper tasks. In this case, the trigger pairs are constructed from a large newspaper corpus and the test data consists of some articles read aloud. Spoken language is very different from written language, but like the latter, the former has also many long-distance dependencies that we want to capture and conversations are also centered in a topic in many cases. Therefore, the trigger-based language model could be used to capture long-distance topic constraints in these conversational speech tasks.

When transcribing conversational speech, however, we find two serious problems for statistical language modeling: the appearance of disfluencies in speech and the small amount of available in-domain data.

Disfluencies can be of different types such as filled pauses (e.g. "uh", "um"), repetitions (e.g. "I I mean"), and repairs (e.g. "he she doesn't like it"). Disfluencies can be considered as noise in the linguistic channel, and they are a serious problem for language modeling, first because sentences can become ungrammatical, for example by having several subjects or by ending unexpectedly; second because disfluencies contribute to data sparseness, since

we could partition the word history into unnecessary equivalence classes (e.g. the trigram model would have different equivalence classes for "my former job" and for "my former uh job"); and third because disfluencies can make the dependencies between words longer, for example when fillers occur between two related words.

The second problem is the small amount of available in-domain data. Contrary to written style text, there are much less available training data for conversational speech domains, because it is much more expensive to produce these corpora than those from newspapers or newswires, for example. The available conversational corpora are usually insufficient to derive a stand-alone task-dependent language model from them. Recently, the number of works that use the World Wide Web (WWW) as a source for extracting training data for language models for conversational speech tasks has been increasing. However, the extracted web pages are not domain matched, and they must be filtered to discard out-of-domain text.

The trigger-based language model is insensitive to disfluencies in speech, because it focuses on the co-occurrence of topic keywords, so it is not affected by the first problem. As for the second problem, the proposed approach uses the available in-domain data to adapt the language model to the conversational speech task.

2.3.3 Adaptation scheme

In this study, the language model adaptation scheme is based on the trigger pairs that are extracted from the available conversational in-domain data. By extracting the trigger pairs from the in-domain data, contrary to the conventional trigger-based language model that constructs the trigger pairs from a general large corpus, we can obtain task-dependent trigger pairs that match more closely the addressed task. In addition, since the probability estimates derived from the target domain might not be reliable, because the amount of in-domain data is typically small, a back-off scheme that uses the statistics from a large corpus is also proposed. In this thesis we propose two different adaptation schemes, which will be presented in chapters 3 and 4.

In the adaptation scheme presented in chapter 3, the trigger pairs are extracted from the target corpus, and their probabilities are estimated from both the in-domain data and the large corpus, resulting in two different sets of trigger pairs, depending on where the probabilities have been estimated from. We apply this method to two different domains: a travel expressions task and an extemporaneous speeches task. Both tasks have the same amount of in-domain training data (3.5M words), which we presume sufficient to extract task-dependent trigger pairs. However, this amount of data might be insufficient to derive reliable probability estimates for the trigger pairs. Therefore, we propose a back-off scheme that backs off to the set of trigger pairs whose probabilities are estimated from the large corpus when there are no applicable trigger pairs in the trigger set estimated from the task corpus.

In chapter 4, the target task is the transcription of panel discussions. In this case, the total size of the available in-domain data is only 134K words, a much smaller size than that of the tasks addressed in chapter 3. All these data are used as the test set, so actually there are no available training data. Therefore, we cannot use the previous adaptation scheme here, because this amount of data is too scarce to obtain the expected quantity and quality of trigger pairs. Instead, we present an adaptation scheme where the trigger pairs are extracted and their probabilities estimated from the initial speech recognition

```
REF:  Parent's education level affects    academic success
HYP:  Further  education       affects an academic success
EVAL:    S         C       D      C    I     C        C
```

Figure 2.2: Example of alignment of hypothesis with reference transcription.

results and from a large corpus, resulting in two different trigger sets, each constructed from a different source. Here, we also use a back-off scheme to back off to the probabilities from the trigger set constructed from the large corpus when there are no applicable trigger pairs in the trigger set constructed from the initial transcription. In addition, we describe the application of the proposed method to another meeting transcription task.

2.4 Language model evaluation measures

2.4.1 Word error rate

The ultimate evaluation measure of a language model is the one that assesses its performance in the particular task for what it was created. In ASR, this measure is the word error rate (WER). The WER is the rate of erroneous words in the output of the speech recognizer. Given a reference (correct) transcription and the output of the ASR system, we align the hypotheses of the output with their respective correct counterparts, and then we count the number and type of errors.

There are three different types of errors: substitutions, insertions, and deletions. When a word is misrecognized and a different one is output instead, it is a substitution (S). If a word appears in a hypothesis but it does not appear in the corresponding acoustic signal, or it is not a misrecognition of any of the words in this acoustic signal, then it is an insertion (I). Deletion (D) is the case when a word is skipped during the recognition, that is, it appears in the observed acoustic signal but it is neither correctly nor incorrectly recognized; it simply does not appear in the hypothesis.

The WER is defined as follows:

$$WER = \frac{\#\text{ of errors}}{\#\text{ of tokens in the reference transcription}} = \frac{S + I + D}{N} \times 100 \qquad (2.20)$$

Figure 2.2 shows the alignment of two sentences with their corresponding errors. The WER here is $WER = 3/6 = 50\%$.

In order to compare the performance of two difference language models with the WER, the acoustic model must be fixed and the WER of the system using the two language models must be compared.

In practice, this measure is not necessarily perfect. In order to reliably measure the WER, we need to perform recognition experiments with large amounts of test data, which consumes a great deal of time. Furthermore, the WER depends on complex interactions among many components, so it is virtually impossible to find analytical expressions for the relationship between the WER and the values of language model parameters.

2.4.2 Perplexity

An alternative measure to WER for evaluating the performance of language models is the *perplexity* [30]. The perplexity can be interpreted as the branching factor of a language model, that is the average number of words that will follow a given word history.

Mathematically, the perplexity is derived from the *entropy*. Let $P(x)$ be the real probability distribution of x and $P_M(x)$ be the probability estimate of x based on language model M. The entropy of $P(x)$ is defined as:

$$H(P) = -\sum_x P(x) \log_2 P(x) \tag{2.21}$$

Then, the *cross-entropy* (also called the *logprob*) of P and P_M is:

$$H(P; P_M) = -\sum_x P(x) \log_2 P_M(x) \tag{2.22}$$

The cross-entropy measures the similarity between the distributions P and P_M. The smaller the cross-entropy, the better the language model M approximates P.

If the size of the test text N is sufficiently large and the language source is ergodic (i.e. every sufficiently long sentence is equally representative), the previous equation can be approximated by:

$$H(P_M) \approx -\frac{1}{N} \sum_{i=1}^{N} \log_2 P_M(w_i | w_1, \cdots w_{i-1}) \tag{2.23}$$

The perplexity of the text with respect to the model M is finally defined as:

$$PPL(P_M) = 2^{H(P_M)} \tag{2.24}$$

The perplexity depends on both the quality of the language model and the complexity of the text. For the same text, the model with the lowest perplexity is the better model, whereas, for the same language model, the text with the highest perplexity is the most difficult to process. Therefore, a comparison between language models must be made with respect to the same text, and also the same vocabulary, because smaller vocabularies result in lower perplexities.

Perplexity is practically a faster way of evaluating the performance of a language model, but it does not take into account acoustic confusability. Moreover, although WER and perplexity are well correlated, lower perplexities do not necessarily imply lower WERs, specially when the reduction in perplexity is low.

2.5 Handling long-distance language models in ASR

2.5.1 The decoder

The decoder is the subsystem of the ASR system that, using the acoustic and language models, searches for the word string that best matches the input feature vectors. Since the search space consists of all the possible combinations of word strings, it is necessary to find some way of reducing the size of this space to make the search feasible. There are two common techniques to achieve this: *path pruning* and *path merging*.

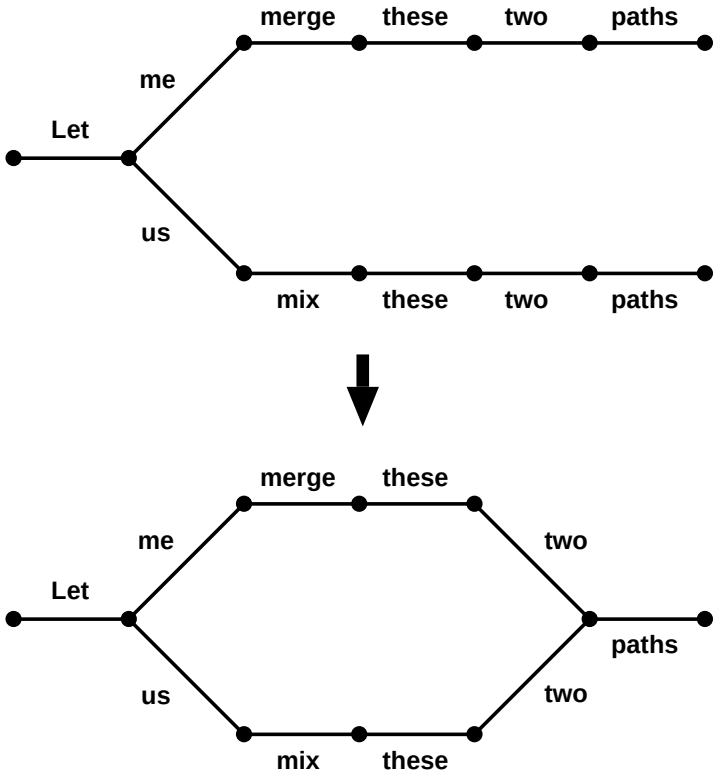

Figure 2.3: Example of path merging when using a trigram language model.

Path pruning is the method that discards very unlikely paths at a particular time point. By using this method, the search space can be considerably reduced, but it is possible that a path very unlikely at an early stage becomes more likely later on, so it may happen that the best path is pruned from the space, thus leading to a search error.

Path merging is the technique that merges two or more converging paths at some point and continue the search with only the more likely of the paths, since a path that is less likely at the point where the paths converge will remain less likely. In order to do this, the converging paths must have equivalent histories according to the language model being used. For instance, if a trigram model is used, two paths can be merged if the final two words in the paths are the same. Figure 2.3 shows an example of path merging with a trigram model. Long-distance language models cause problems to this decoding scheme, because it is not possible to merge paths so frequently, and therefore the search space will probably remain too large.

Moreover, long-distance language models usually require much more memory than a standard n-gram model, which added to the memory required by the acoustic model can make it very costly to use these language models during the decoding step.

In order to overcome these problems, there are two alternative methods for integrating long-distance language models into the speech recognition framework, namely *N-best* and *word graph rescoring*. In these methods speech recognition takes place in two or more passes. The first pass generally uses a simple language model such as a bigram to generate a simplified search space. It is hoped that the best hypothesis according to the final language model is not pruned from this space at this step. The output from

the first pass usually takes the form of a *word graph*, which is then rescored with a more complex language model (typically a trigram or 4-gram) to obtain the final results. These results consist of an *N-best list* of hypotheses, which can in turn be rescored with a more complex language model. These approaches have also the advantage that experiments for evaluating different language models can be performed without the need of computing the acoustic scores again.

2.5.2 N-best rescoring

An N-best list is the list of the N most likely hypotheses output by a speech recognizer. It usually contains for each hypothesis its acoustic score and the language score for each word.

In the N-best rescoring method, each of the hypotheses in the N-best list is rescored based on a combination of the scores provided by the speech recognizer and the new language model probabilities provided by an alternative (generally more complex) language model. Then, the hypotheses are reordered based on the new scores and the most likely hypothesis is presented as the output of the whole recognition process.

N-best rescoring is widely used in language modeling for ASR, because of its easy implementation and fast evaluation, since only several hundred hypothesis are typically considered for each sentence. However, as the last possible step for applying a language model, the search space is much smaller than in previous steps, so the correct hypothesis is more likely to have already been pruned.

2.5.3 Word graph rescoring

A word graph or *lattice* is a directed acyclic graph which contains the paths that were considered more likely by the acoustic and language models during the initial decoding pass. Each of the nodes corresponds to a hypothesized word boundary, and is associated with a time, while each arc is labeled with a word and its corresponding acoustic and language model scores.

Analogously to the N-best rescoring method, in the word graph rescoring technique the language model scores can be replaced by those from an alternative language model. Then, we can search through the lattice for the path that is now considered the most likely.

This approach is preferred to the previous one, even though it is a little more computationally expensive than N-best rescoring, because the search space is not as constrained in this case, so we have more chances of finding the correct hypothesis.

Chapter 3

Trigger-Based Language Model Construction by Combining Different Corpora

3.1 Introduction

This chapter presents a novel trigger-based language model adaptation scheme for the transcription of two different conversational speech tasks that takes advantage of two distinct corpora.

When training the trigger-based language model, we usually find a fundamental problem, depending on the nature of the training data. When the trigger pairs are trained from a large corpus, many of the pairs are not task-dependent, because the corpus is usually too general. Therefore, the effectiveness of the trigger-based language model is undermined by the specificity of the target task. On the other hand, when the training data set is from the same domain as the target task, its size is usually insufficient and the probability estimates are unreliable.

To overcome this trade-off between generality and sparseness, we propose an approach that takes advantage of two different corpora to create a trigger-based language model so that the trigger pairs are dependent on the target task and have reliable estimates.

The rest of this chapter is organized as follows. Section 3.2 describes the proposed approach in detail. In subsection 3.2.1 the method for extracting the trigger pairs from the task corpus based on two different measures is explained. Then, the probability estimation of the trigger pairs from the two different corpora is discussed in subsection 3.2.2, and the proposed language model is formulated in subsection 3.2.3. Section 3.3 discusses the application of the proposed approach to a travel expressions task and evaluation in terms of perplexity and speech recognition accuracy, while section 3.4 deals with the application and evaluation in an extemporaneous speeches task.

3.2 Proposed approach

Figure 3.1 illustrates the outline of the proposed approach. First, the trigger pairs are extracted from a text corpus that matches the target task (task corpus). Then the probabilities of the pairs are estimated, based on their co-occurrence frequency within a text

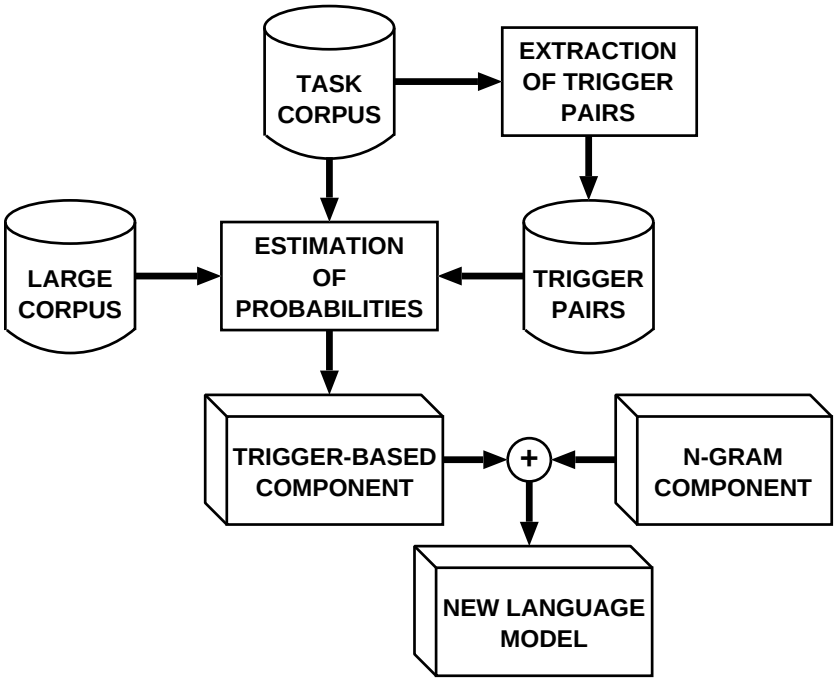

Figure 3.1: Outline of the proposed approach.

window, from two different corpora: the mentioned task corpus and a large text corpus, providing us with two different sets of trigger pairs with their corresponding probabilities. Finally, the resulting trigger-based component is combined with the n-gram component to produce a new language model.

The proposed model uses a back-off scheme that uses a combination of the probabilities from the two trigger pair sets when the trigger pairs can be found in the set trained from the task corpus. Otherwise, the probabilities from the set trained from the large corpus are used.

By extracting the trigger pairs from the target domain, we solve the generality problem, while we avoid the data sparseness problem by using the set of trigger pairs whose probabilities are estimated from the large text corpus.

3.2.1 Extraction of trigger pairs from task corpus

A trigger pair is a pair of content words that are semantically related to each other. Trigger pairs can be represented as $A \rightarrow B$, which means that the occurrence of A triggers the appearance of B, that is, if A appears in a document, it is likely that B will come up afterwards.

The trigger pairs are first extracted from a text corpus that matches the target domain. In this way, we can obtain task-dependent trigger pairs. For the selection of pairs, instead of the average mutual information used in [47, 56], we adopt two different criteria: the term frequency/inverse document frequency (TF/IDF) measure [59] and the log likelihood ratio [19]. We use the former for preliminary experiments because of its simplicity, while the latter, although more computationally demanding, is used for its powerfulness.

Extraction based on the TF/IDF measure

The TF/IDF value of a term t_k in a document D_i is computed as follows:

$$v_{ik} = \frac{tf_{ik} \log (N/df_k)}{\sqrt{\sum_{j=1}^{T} (tf_{ij})^2 [\log(N/df_j)]^2}} \tag{3.1}$$

where tf_{ik} is the frequency of occurrence of t_k in D_i, N is the total number of documents, df_k is the number of documents that contain t_k, and T is the number of terms in D_i.

For each document, we create all possible word pairs, including pairs of the same words (self-triggers), with the base forms and parts of speech (POS) of all the words with a TF/IDF value above a threshold. POS-based filtering is introduced to discard function words, as well as a word stop list to ignore words of very frequent appearance. By using base forms we avoid same-stem triggers (trigger pairs whose component words have the same stem but different inflection), and we can apply the trigger pair when a word is presented with any inflected form. For example, in the sentences *terebi wo miru* (I watch television) and *terebi wo mita* (I watched television), it seems reasonable that the correlation between *terebi* (television) and *miru* (to watch) should be used in both cases. In addition, by using the POS information we distinguish between homonyms with different POS when applying the trigger pairs. For instance, *kaeru* (frog) should have a higher probability of triggering *ike* (pond) when it is a noun than when it is a verb, in which case its meaning is "to go back".

Extraction based on the log likelihood ratio

Given a contingency table with the frequency of the following co-occurrence pairs:

$$\begin{array}{ll} a) A + B & c) \neg A + B \\ b) A + \neg B & d) \neg A + \neg B \end{array}$$

where $A + \neg B$ represents the two pairs $A \rightarrow \neg B$, $\neg B \rightarrow A$ formed by A and any word that is not B, the log likelihood ratio (LLR) of the pair $A \rightarrow B$ is calculated as follows:

$$\begin{aligned} -2 \log \alpha = 2[&a \log a + b \log b + c \log c + d \log d - (a+b) \log(a+b) - (a+c) \log(a+c) \\ &- (b+d) \log(b+d) - (c+d) \log(c+d) + (a+b+c+d) \log(a+b+c+d)] \end{aligned} \tag{3.2}$$

For each document, we first create all possible pairs with the base forms and POS of all the words in it, including self-triggers. Again, POS-based filtering and a stop list are used to remove function words and high frequency words, respectively. Then, we compute the LLR for each pair and choose the trigger pairs with a ratio greater than a threshold.

3.2.2 Probability estimation from two corpora

The probabilities of the trigger pairs are then estimated from two different corpora by using a text window to calculate the co-occurrence frequency of the pairs inside it. This text window consists of the 20 words previous to the one being processed.

The two distinct corpora used are the text corpus that matches the target task and a large text corpus. The probability estimation stage results in two different sets of trigger

pairs: the trigger pairs with the probabilities estimated from the task corpus (hereafter trigger set TC), and the trigger pairs whose probabilities are estimated from the large corpus (hereafter trigger set LC). The trigger set TC provides a probability distribution more faithful to the target domain, whereas the trigger set LC offers a more reliable distribution that can cope with the problem of data sparseness that we discussed above.

The probability of each trigger pair $w_1 \rightarrow w_2$ is computed as follows:

$$P_{TP}^{IT}(w_2|w_1) = \frac{N(w_1, w_2)}{\sum_j N(w_1, w_j)} \tag{3.3}$$

where $N(w_1, w_2)$ denotes the number of times the words w_1 and w_2 co-occur within the text window, and j runs throughout all words triggered by w_1.

3.2.3 Proposed trigger-based language model

The proposed trigger-based language model is then constructed by linearly interpolating the probabilities of the trigger pairs with those of the baseline trigram (3-gram) model, so that both long and short-distance dependencies can be captured at the same time.

The probability of the new language model for a word w_i given the word history $H = w_{i-L}, \cdots w_{i-1} \triangleq w_{i-L}^{i-1}$ is computed in the following way:

$$P_{LM}(w_i|H) = \frac{1}{L} \sum_{j=i-L}^{i-1} P_{LM}(w_i|w_j) \tag{3.4}$$

$$P_{LM}(w_i|w_j) = \begin{cases} P_{NG}(w_i|w_{i-n+1}^{i-1}), \text{ if } P_{TP}^{IT}(w_k|w_j) = 0, P_{TP}^{LC}(w_l|w_j) = 0, \forall k, l \\ \lambda P_{NG}(w_i|w_{i-n+1}^{i-1}) + (1-\lambda) P_{TP}^{LC}(w_i|w_j), \text{ if } P_{TP}^{IT}(w_k|w_j) = 0, \forall k \\ \lambda P_{NG}(w_i|w_{i-n+1}^{i-1}) + (1-\lambda)\big(\delta P_{TP}^{LC}(w_i|w_j) + (1-\delta) P_{TP}^{IT}(w_i|w_j)\big), \text{otherwise} \end{cases} \tag{3.5}$$

Here L is the number of words in the history H; P_{NG} is the probability of the n-gram component; P_{TP}^{TC} is the probability of the trigger set TC; P_{TP}^{LC} is the probability of the trigger set LC; λ is the language model interpolation weight; and δ is the trigger set interpolation weight.

When there are no words triggered by h in either of the two sets, the trigram model alone is used. When there are no trigger pairs for h in the trigger set TC, the trigram probabilities and the probabilities from the trigger set LC are linearly interpolated. Otherwise, the probabilities of the trigram are linearly interpolated with a linear interpolation between the probabilities from both trigger sets.

3.2.4 N-best rescoring

The new language model is used to rescore the N-best hypotheses output by a baseline ASR system. The system provides us with acoustic and language model scores for each of the words in every hypothesis.

Words in each hypothesis are added in order to a word history buffer, which is cleared when the hypothesis processing is over. The language model score for each hypothesis is updated by using this buffer and the previous equations. The hypothesis with the highest new total score is regarded as the new 1-best sentence.

The number of trigger pairs used during the rescoring process is limited to be those with a probability above a threshold.

Table 3.1: Example of trigger pairs extracted from the BTEC.

Triggering word	Triggered word
tounyoubyou (diabetes)	*menyuu* (menu)
tounyoubyou (diabetes)	*kanja* (patient)
sensei (doctor)	*miru* (to examine)
kenpou (constitution)	*sengo* (postwar)
guragura (loose)	*ha* (tooth)
koon (cone)	*aisukuriimu* (ice cream)
koukoku (advertisement)	*kouka* (effect)
susume (recommendation)	*wain* (wine)
tai (Thailand)	*shoo* (show)
tegami (letter)	*ate* (addressed to)
nimotsu (baggage)	*orosu* (to unload)
kutsu (shoe)	*uriba* (selling area)
teeburu (table)	*katazukeru* (to tidy up)

3.3 Evaluation in travel expressions task

3.3.1 Corpora and procedure

The Basic Travel Expression Corpus (BTEC) [71] is a conversational text corpus consisting of sentences from many different topics that usually appear in travel conversations. It is divided in two disjoint sets: training and evaluation. The former contains 467,964 utterances and 3.5 million Japanese morphemes (hereafter words), and the latter comprises 24,682 utterances and 184 thousand words.

The trigger pairs were extracted from the Japanese version of the BTEC. We had to use the utterance as the document unit, since utterances in this corpus are not related to each other.

The threshold for the TF/IDF value was chosen to be 0.2 so that the hit rate in the evaluation corpus of the trigger pairs created with only the POS-based filtering were 20%, and was empirically tuned later to produce a threshold of 0.15.

The threshold for the LLR was initially chosen to be 10 based on a subjective judgment of the goodness of the pairs from a sample taken at random, and it was later tuned empirically, producing the value 2. The hit rate in the evaluation corpus of the trigger pairs created by using only the POS-based filtering was 19%.

Table 3.1 shows some examples of trigger pairs extracted from the BTEC that were actually used in the experiments. A bigger list can be found in appendix A.

The probabilities of the trigger pairs were estimated from two different corpora: the Mainichi Shimbun newspaper corpus and a conversational text corpus extracted from the WWW [34] (hereafter web corpus). We used 5 years (1991-1995) of articles from the Mainichi Shimbun corpus, consisting of 130 million words. The web corpus consists of conversational texts that can be found in the WWW, such as chat logs, and comprises 270 million words, of which we used 122 million words.

Figure 3.2 illustrates a sample from the web corpus. Lines starting with '#' are

S-ID:tsubame01-11-133-244 URL:kataribe.com:80/BBS/line/040.html 部分削除:5:（汗）
いつ帰ろう１６日の大文字焼までに帰ればいいのだ（ ^^; ＞帰還日　後は滞在費（核爆）
S-ID:tsubame01-11-133-245 URL:kataribe.com:80/BBS/line/040.html
こちらも猫だらけです＜葛飾猫トラップ多数。
S-ID:tsubame01-11-133-246 URL:kataribe.com:80/BBS/line/040.html
特に今年生まれた三つ子は反則だぁ（Ｔ＾Ｔｏ
S-ID:tsubame01-11-133-247 URL:kataribe.com:80/BBS/line/040.html
そこが問題だぁね＞滞在費　ビジネスホテルっすか？
S-ID:tsubame01-11-133-248 URL:kataribe.com:80/BBS/line/040.html
東京の下町も一度見てみたいです。
S-ID:tsubame01-11-133-249 URL:kataribe.com:80/BBS/line/040.html
行ったことないので（ ^^;
S-ID:tsubame01-11-133-250 URL:kataribe.com:80/BBS/line/040.html
取り合えず、はりにゃんのお部屋に泊めてもらう（ ^^; ＞９日　それ以後は不明（汗）
S-ID:tsubame01-11-133-251 URL:kataribe.com:80/BBS/line/040.html
やせっぽっちで片足のない老猫がいるので、ついつい食事を分けてやってしまうにゃぁ。
S-ID:tsubame01-11-133-252 URL:kataribe.com:80/BBS/line/040.html
寝袋も持って行くので、どこかで野宿の可能性もあるにゃ（ ^^; ＞宿泊場所
S-ID:tsubame01-11-133-253 URL:kataribe.com:80/BBS/line/040.html
なんだかなぁ（ ^^;　とりあえず掲示板汎用の方で、１３日に下町散策が走るやも、きっとぽてぽて歩いて甘味所に入るを繰り返すつあーさ（ ^ー^
S-ID:tsubame01-11-133-254 URL:kataribe.com:80/BBS/line/040.html 部分削除:9:（火暴） 部分削除:39:（火暴）
１４日にしようよぅ＞下町オフ＜　猛烈にがそばれば休めるかもしれないから
S-ID:tsubame01-11-133-255 URL:kataribe.com:80/BBS/line/040.html 部分削除:26:（笑）
不観樹さん、うちのスゴイ部屋だったら泊めてあげますよ
S-ID:tsubame01-11-133-256 URL:kataribe.com:80/BBS/line/040.html 部分削除:20:（＾ー＾）
ＩＣＱで相談の結果泊ることになりそうにゃ
S-ID:tsubame01-11-133-257 URL:kataribe.com:80/BBS/line/040.html
明日はバイトだから早く寝るにゃ
S-ID:tsubame01-11-133-258 URL:kataribe.com:80/BBS/line/040.html
皆さんお休みなのにゃ
S-ID:tsubame01-11-133-259 URL:kataribe.com:80/BBS/line/040.html 部分削除:18:（笑） 部分削除:36:（笑）
明日バイトだけど、もうこんな時間にゃでも昨日より早いからいいのにゃ
S-ID:tsubame01-11-133-260 URL:kataribe.com:80/BBS/line/040.html
うぃーす、お久しぶり★
S-ID:tsubame01-11-133-263 URL:kataribe.com:80/BBS/line/040.html
Ｊｏｓｈｉｎが全店休みになりますんで
S-ID:tsubame01-11-133-264 URL:kataribe.com:80/BBS/line/040.html
とりあえずのんびりします…東京下町かぁ、行きたいなぁ
S-ID:tsubame01-11-133-267 URL:kataribe.com:80/BBS/line/040.html
って、既に昼ではないか。
S-ID:tsubame01-11-133-268 URL:kataribe.com:80/BBS/line/040.html 部分削除:21:（爆）
クリエイタと酒飲み連中にはよくあることだが

Figure 3.2: A sample from the web corpus.

Table 3.2: Experimental setup for the application of the proposed approach to the BTEC.

Test set	1524 utterances (11K words)
ASR system	ATRIUMS 2.2
Baseline language model	BTEC + Mainichi trigrams
Vocabulary	36K words
OOV rate	0.19%
Number of hypotheses (N)	100
Baseline word accuracy	87.64%
Oracle word accuracy	94.53%
Baseline perplexity	16
Stop word list threshold	500, 1000, 2000, 3000, 5000

sentence IDs. It can be seen that some sentences include non-lexical information such as emoticons (e.g. " (^^; ", " (^ — ^ "), special characters (e.g. "★"), and other emotional markers (e.g. " (汗) ", meaning "sweat"). These items were removed in a preprocessing step.

3.3.2 Experimental setup

The ASR system ATRIUMS 2.2 [69] was used to output the N-best lists. The size of the vocabulary was 36K words. This system normally uses a bigram model in a first stage and a trigram afterwards, in an optional rescoring stage. The BTEC bigram was used in the first recognition stage, and a linear interpolation between the BTEC and Mainichi trigrams, with interpolation weights of 0.99 and 0.01, respectively, was used for the second stage. The test set consisted of 1524 utterances (11K words) taken from the BTEC evaluation corpus (sets 1, 2 and 3) and the number of output hypotheses N was 100. The baseline perplexity was 16.

We obtained an average word recognition accuracy of 87.64% with this baseline language model, and the maximum average recognition accuracy that could be attained by choosing the best hypothesis from the N-best each time (oracle word recognition accuracy) was 94.53%.

The experimental setup is summarized in table 3.2.

3.3.3 Perplexity evaluation

We evaluated the perplexity of the proposed language model for different values of the hit rate of the trigger pairs in the test set, determined by the threshold for the frequency of the words in the stop list. The values for this threshold were 500, 1000, 2000, 3000, and 5000. We compared the perplexity of the model constructed from both the BTEC and the web corpus, the model built from the BTEC and the Mainichi Shimbun corpus, and the one that used only the BTEC, both to extract the trigger pairs and to calculate their probabilities. We compared these three models for each of the two criteria used for the extraction of the trigger pairs. For the TF/IDF measure, the number of extracted trigger pairs varied from 447,060 to 1,052,342 for the first model, from 418,629 to 976,656 for the

Table 3.3: Topics used in CSJ.

#	Broad topic	Number of files
0	(Not specified)	222
1	Joyful memory of my life	137
2	Sad memory of my life	134
3	The town I live in	134
4	This is what I'm interested in	151
5	Impressive event of my life	167
6	Commentary on recent news	152
7	If I go to an isolated island, I will bring...	101
8	How to make...	151
9	History of...	100
10	My most precious thing/people	100
11	Things that I want to endow for the 21st century	150

second one, and from 325,253 to 880,957 for the third one. For the LLR, the number of extracted trigger pairs varied from 412,678 to 821,093 for the first model, from 388,912 to 767,157 for the second one, and from 300,849 to 668,878 for the third one.

The two criteria gave similar results, and figures 3.3 and 3.4 show the results when we used the TF/IDF and the LLR criterion, respectively. We can see that the perplexity did not change significantly in any of the cases. One of the possible reasons for this is that, since the utterances of BTEC are unrelated to each other, we could not use the information of the previous sentences for our trigger-based language model. Furthermore, most utterances in BTEC are short, so it is difficult to extract good trigger pairs from them.

3.3.4 Rescoring experiments

We then carried out rescoring experiments with the output of the baseline system. We compared the word recognition accuracy of the models constructed from the BTEC and the web corpus, the BTEC and the Mainichi Shimbun corpus, and only the BTEC, for each of the two extraction criteria.

Figures 3.5 and 3.6 show these results. The WER is plotted against the hit rate of the trigger pairs in the test set. The best word recognition accuracy obtained was 87.71%, that is, we achieved a global 0.07% improvement when we used trigger pairs based on the LLR, a stop list threshold of 5000, and the probabilities were computed from the web corpus.

3.4 Evaluation in extemporaneous speeches task

3.4.1 Corpora and procedure

The Corpus of Spontaneous Japanese (CSJ) [49] is a conversational corpus consisting of lectures on various academic topics and extemporaneous speeches about different matters

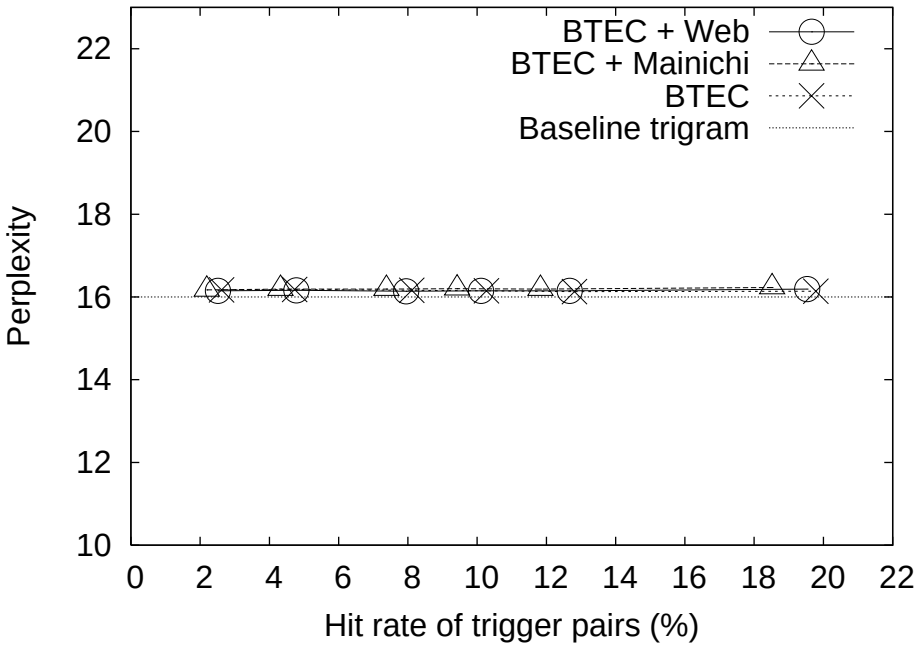

Figure 3.3: Perplexity against hit rate of trigger-based models for different sets of trigger pairs extracted from the BTEC with the TF/IDF measure.

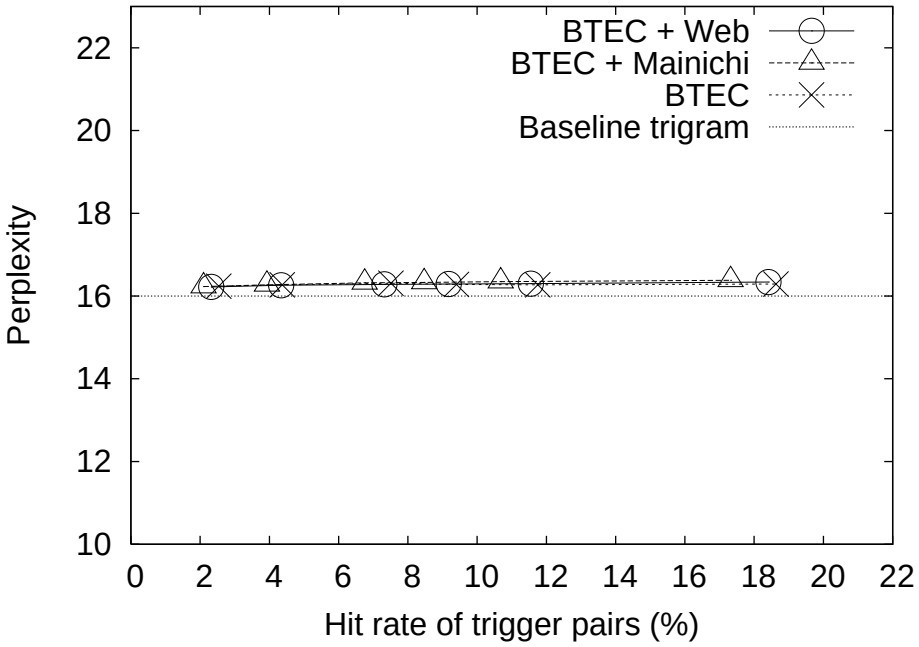

Figure 3.4: Perplexity against hit rate of trigger-based models for different sets of trigger pairs extracted from the BTEC with the LLR.

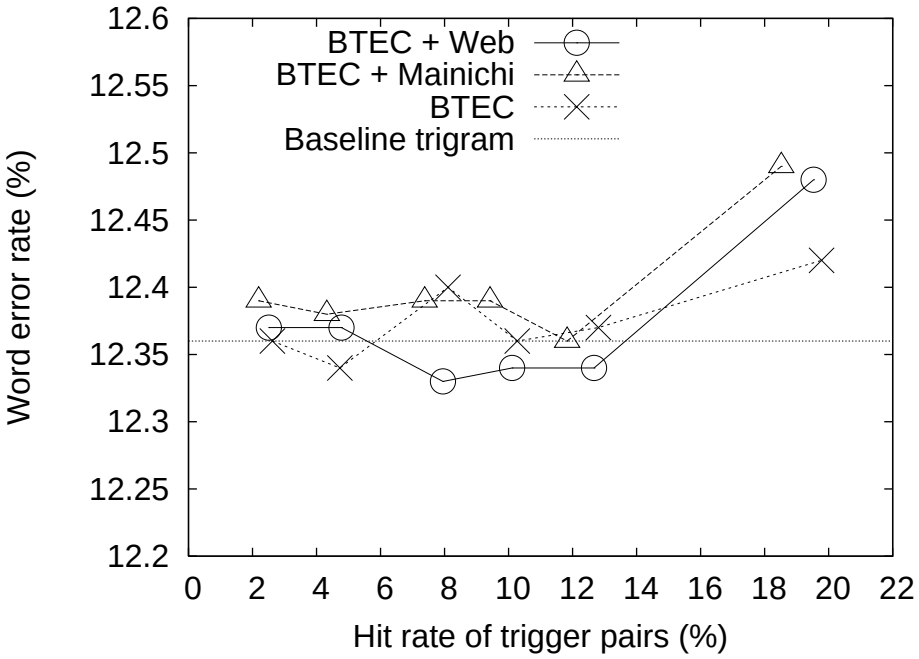

Figure 3.5: Word error rate against hit rate of trigger-based models for different sets of trigger pairs extracted from the BTEC with the TF/IDF measure.

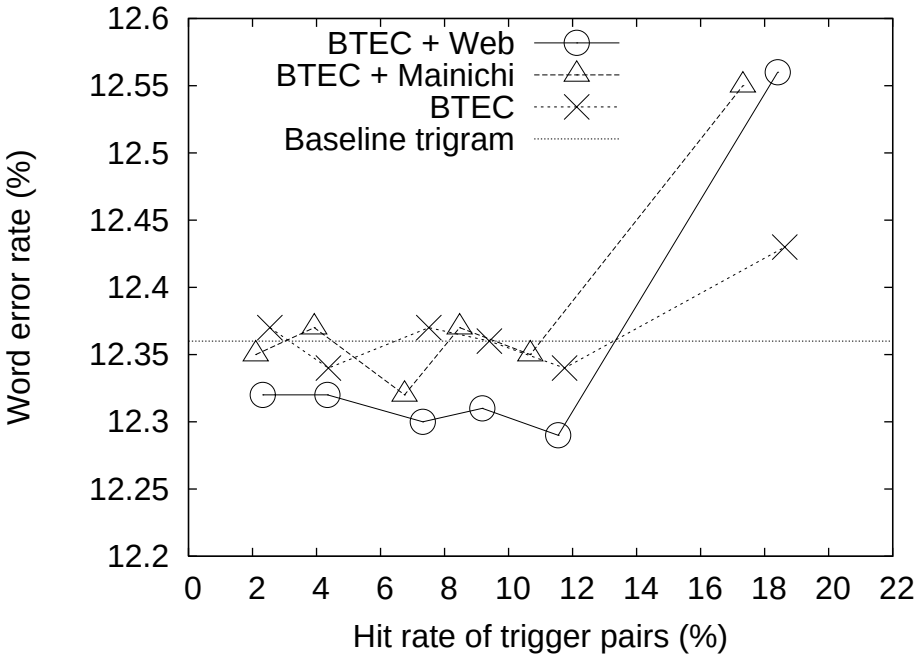

Figure 3.6: Word error rate against hit rate of trigger-based models for different sets of trigger pairs extracted from the BTEC with the LLR.

Table 3.4: Example of trigger pairs extracted from the CSJ.

Triggering word	Triggered word
machi (town)	*sumu* (to live)
oya (parent)	*kodomo* (child)
mujintou (desert island)	*shima* (island)
hontouni (really)	*sugoi* (amazing)
haha (mom)	*chichi* (dad)
nihon (Japan)	*amerika* (America)
taberu (to eat)	*oishii* (delicious)
shigoto (job)	*kaisha* (company)
nihonjin (Japanese)	*nihon* (Japan)
ryokou (travel)	*kaigai* (abroad)
sensei (teacher)	*gakkou* (school)
byouin (hospital)	*nyuuin* (hospitalization)
daigaku (university)	*koukou* (high school)

Table 3.5: Specification of used corpora.

Corpus name	Contents	Type of language	Size
CSJ	Extemporaneous speeches	Spoken language	3.5M words
Mainichi Shimbun	Newspaper articles	Written language	289M words
Web corpus	Chat logs	Spoken language	270M words

such as hobby and travel. We used the extemporaneous speeches, which are 10 to 12 minutes monologues on diverse topics from a list of 12 (table 3.3). The extemporaneous speeches are divided into 1705 speeches of training data, comprising 3.5 million words, and 10 speeches of evaluation data, containing 18 thousand words.

The trigger pairs were extracted from the CSJ training data. We used the lecture as the document unit. The threshold for the TF/IDF value was initially chosen to be 0.015 based on a subjective judgment of the goodness of the pairs from a sample taken at random, and it was later tuned empirically, producing the value 0.031.

Table 3.4 shows some examples of trigger pairs extracted from the CSJ that were actually used in the experiments. A bigger list can be found in appendix A.

For estimating the probabilities, we used two different corpora: the Mainichi Shimbun newspaper corpus and the web corpus. We used 11 years (1991-2001) of articles from the Mainichi Shimbun corpus, consisting of 289 million words. The whole 270 million words from the web corpus were used. Being conversational, the web corpus is closer in style to the CSJ than the Mainichi Shimbun newspaper corpus, so we expected to get better experimental results with the former. Table 3.5 summarizes the corpora used in this work.

The language model interpolation weight and the trigger set interpolation weight were empirically tuned to produce the values 0.7 and 0.76, respectively.

Table 3.6: Experimental setup for the application of the proposed approach to the CSJ.

Test set	10 speeches (18K words)
ASR system	Julius 3.4.2
Baseline language model	CSJ back-off trigram
Vocabulary	30K words
OOV rate	0.62%
Number of hypotheses (N)	100
Baseline word accuracy	66.76%
Baseline perplexity	74
Stop word list threshold (CSJ)	500, 1000, 2000, 3000, 5000, none
Stop word list threshold (Mainichi Shimbun)	100000, 200000, 400000, none

3.4.2 Experimental setup

For the CSJ experiments, we used the ASR system Julius 3.4.2 [48]. The size of the vocabulary was 30K words. We created a word bigram and a back-off trigram from the CSJ training corpus, and we used the CSJ test set for the experiments. The number of output hypotheses N was also 100 here. The average word recognition accuracy was 66.76% with this baseline language model. The baseline perplexity in this case was 74.

The experimental setup is summarized in table 3.6.

3.4.3 Perplexity evaluation

We evaluated the test-set perplexity by the proposed language model for different values of the hit rate of the trigger pairs in the evaluation data, determined by the threshold for the frequency of the words in the stop list. We compared four different models: the model that was constructed by using the CSJ and the web corpus (CSJ + Web), the model constructed with the CSJ and the Mainichi Shimbun corpus (CSJ + Mainichi), a model that used only the CSJ corpus (CSJ), both to extract the trigger pairs and to calculate their probabilities, and a model that used only the Mainichi Shimbun corpus (Mainichi), extracting the trigger pairs from the portion corresponding to year 2001 and estimating their probabilities from the whole corpus. We did not create a model only from the web corpus because it is not divided into documents, so it is not suitable for the TF/IDF computation.

The values of the threshold for the stop list were 500, 1000, 2000, 3000, 5000, and no stop list, for the first three mentioned models, and 100000, 200000, 400000 and no stop list, for the last one.

The number of extracted trigger pairs varied from 11,483,557 to 12,048,275 for the CSJ + Web model, from 11,109,675 to 11,804,186 for the CSJ + Mainichi model, from 3,838,096 to 3,907,486 for the CSJ model, and from 22,774,387 to 23,810,712 for the Mainichi model.

The results are illustrated in figure 3.7. The highest perplexity reduction was 12.8%. We can see that the CSJ + Web model and the CSJ + Mainichi model resulted in very similar perplexity results. Furthermore, the perplexity of the models that used two corpora was always lower than that of the models that used only one corpus.

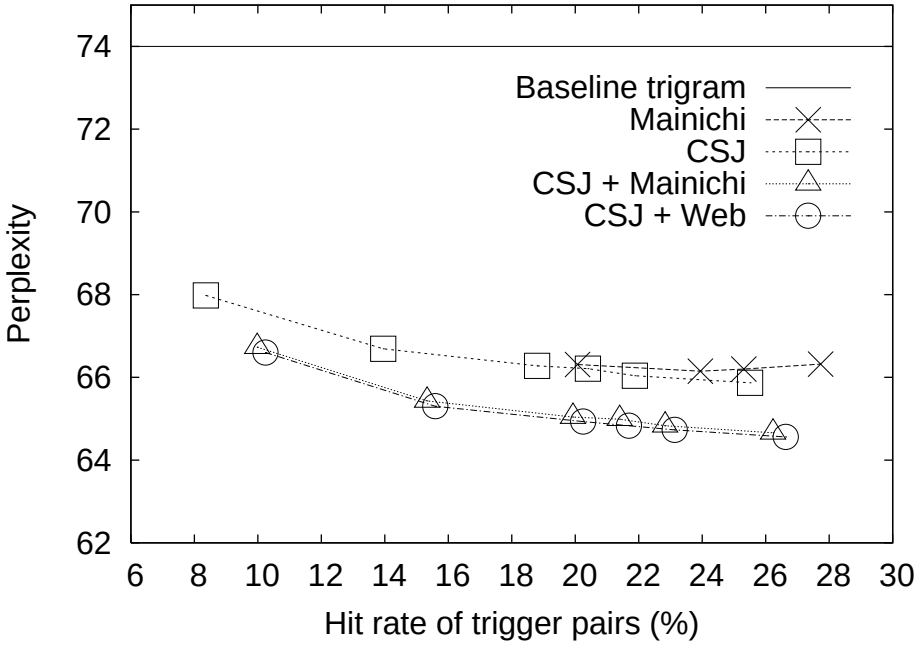

Figure 3.7: Perplexity against hit rate of trigger-based models for different sets of trigger pairs extracted from the CSJ with the TF/IDF measure.

3.4.4 Rescoring experiments

Next, we performed rescoring experiments with the output of the baseline system. We compared the word recognition accuracy of the models constructed from the CSJ and the web corpus, the CSJ and the Mainichi Shimbun corpus, and only the CSJ. The results are shown in figure 3.8. As can be seen, the model that used both the CSJ and the web corpus achieved the lowest error rate. The models that used two corpora performed on average better than the model that used only the CSJ.

3.5 Conclusion

We presented a novel approach to the trigger-based language model based on two different corpora. Generally in language modeling, when the training corpus matches the target task, its size is typically small, and therefore insufficient for providing reliable probability estimates. On the other hand, large corpora are often too general to capture task dependency. The proposed approach tries to overcome this generality-sparseness trade-off problem by taking advantage of the task corpus in order to obtain task-dependent trigger pairs, while a large corpus is used to cope with the data sparseness problem.

A significant improvement in perplexity was achieved when using the two corpora for constructing the model, as compared with the baseline trigram and the models that use only one corpus. This suggests that the proposed method effectively takes advantage of the two different information sources to obtain task-dependent trigger pairs with more reliable probability estimates. In addition, a small improvement in word recognition accuracy was observed during N-best rescoring.

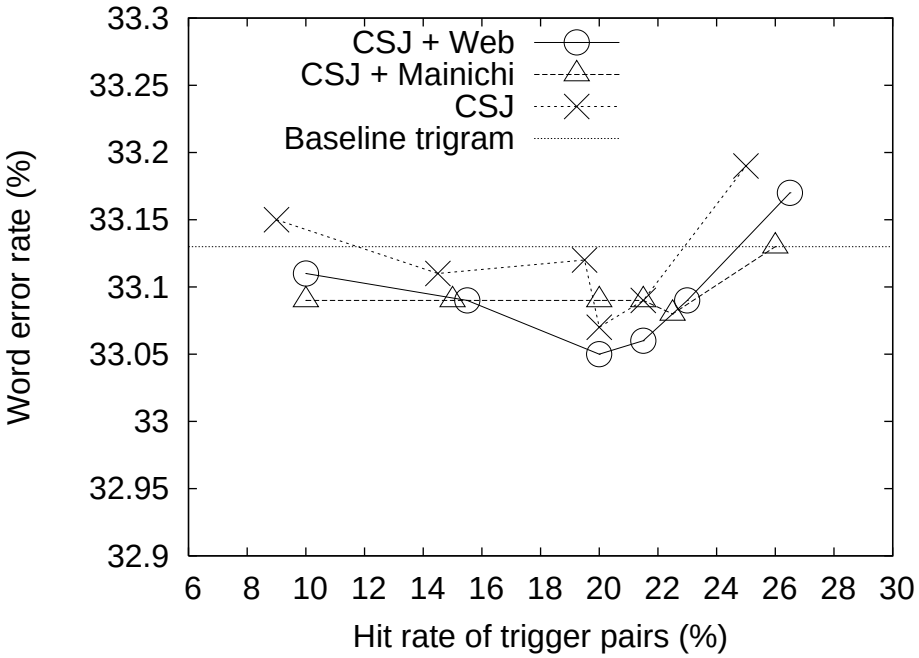

Figure 3.8: Word error rate against hit rate of trigger-based models for different sets of trigger pairs extracted from the CSJ with the TF/IDF measure.

We found out that, contrary to our expectations, the performance of the web corpus was almost identical to that of the Mainichi Shimbun. The corpus size seems to supersede the differences in style.

The proposed approach is particularly useful in tasks where large amounts of training data are not readily available, since we have observed that, with the proposed method, general corpora such as the Mainichi Shimbun can be used to complement small task corpora. The applicability of this method will be revisited in chapter 5.

Chapter 4

Trigger-Based Language Model Adaptation for Automatic Transcription of Meetings

4.1 Introduction

As we have seen in the previous chapters, the conventional trigger-based language model has some limitations. This model has been mostly applied to corpora of newspaper articles. This kind of corpora are usually too general in topic and do not closely match the specific test data. Moreover, it has been observed that much of the potential of trigger-based language models lies in self-triggers. Self-triggers are virtually equivalent to the cache-based language model, so the original trigger-based language model does not significantly outperform the cache-based model.

This chapter addresses an effective implementation of the trigger-based language model mainly targeting at a meeting transcription task to overcome the model's limitations. The transcription of meetings and lectures is one of the promising applications of large vocabulary continuous speech recognition. The subject matter in a meeting is fairly homogeneous during it, so we can expect to find keywords related in their topic throughout the whole session. The trigger-based language model could be used to capture these constraints, but typical large corpora such as newspapers are too general to extract task-specific trigger pairs and their statistics. On the other hand, the data from a single meeting session can be used to extract trigger pairs, and we expect that the probabilities of the trigger pairs can also be estimated from these data.

In the proposed approach, we regard a meeting session as a document unit, and the trigger pairs are extracted from its initial speech recognition results. The initial transcription, though containing errors, can provide useful information about the topic and speaking style of the meeting. We introduce several techniques that filter this useful information from the initial transcription and also exploit a large corpus based on a back-off scheme. The resultant model is used for rescoring the initial speech recognition results.

The rest of this chapter is organized as follows. Section 4.2 describes the task addressed in this work, as well as the proposed approach. Section 4.3 deals with the extraction of trigger pairs from the initial transcription. Then, their probability estimation and an enhancement based on a back-off scheme using a large corpus are explained in section 4.4. The perplexity evaluation of these models in a panel discussion transcription task

Table 4.1: Specification of the "Sunday Discussion" corpus.

ID	# Speakers	# Utterances	# Words	Agenda
0624	5	534	14,423	Reformation of Japanese economy
0805	5	665	15,270	National budget reformation
0819	5	609	14,828	Deflation in Japanese economy
0902	8	541	15,147	Measures against unemployment
0916	6	612	16,128	Terrorism on 9.11
1118	8	474	15,411	Employment and recession
1125	5	371	16,130	Economy stimulus package
1209	5	613	17,150	Budget for the coming year
1216	5	559	14,633	Measures against unemployment
0113	5	524	14,789	Economic prospects of the new year
Average	—	550	15,391	—

is presented in section 4.5, as well as a further enhancement by combining the model with n-gram model adaptation. Speech recognition evaluation in this task is portrayed in section 4.6. Finally, section 4.7 describes the application of the proposed approach to another meeting transcription task.

4.2 Trigger-Based Language Model Adaptation

This section describes the addressed task, the corpora used to create the proposed model, as well as the concept of the proposed approach.

4.2.1 Description of task and corpora

The target task in this work is the transcription of panel discussions from a Japanese TV program called *Sunday Discussion* broadcasted by NHK [1]. This program consists of discussions on current political and economic issues by politicians, economists and other experts in the field. A specific agenda is given for each session of the discussions. A chairperson also takes part and prompts the speakers. The duration of each session is one hour. Ten programs recorded from June 2001 to January 2002 were used in this work. These programs were chosen arbitrarily to cover diverse topics and a sufficient variety of speakers. The average number of utterances and words per program is 550 and 15K, respectively. The total number of words in the test set is 134,405. Figure 4.1 shows the specification of this corpus.

We also make use of a large corpus of the minutes of the National Diet (Congress) of Japan [1] from 1999 to 2002. We selected this corpus because of its similarity in topic with the panel discussion programs, since both corpora mainly deal with politics and economics. The total number of words in the corpus is 71M. Documents in this corpus are divided by the kind and date of meetings, and the total number of documents is 2866. Among them, we select 671 documents from the year 2001 as a portion similar to the test set. Figure 4.2 shows the description of this corpus.

Table 4.2: Categories and number of documents in the National Diet corpus.

Plenary sessions	271
Committees:	
Cabinet	98
Internal Affairs and Communications	171
Judicial Affairs	163
Foreign Affairs	100
Financial Affairs	129
Education, Culture, Sports, Science and Technology	136
Health, Labour and Welfare	177
Agriculture, Forestry and Fisheries	127
Economy, Trade and Industry	132
Land, Infrastructure and Transport	149
Environment	82
Security	60
Fundamental National Policies	30
Budget	189
Audit and Oversight of Administration	81
Rules and Administration	323
Discipline in the House of Representatives	6
Others	442
Total	2,866

4.2.2 Proposed approach

Since each session of the discussions focuses on a particular topic, we expect to find topic-related words during the whole program. In order to capture these long-distance dependencies, we propose to use the trigger-based language model. This model, however, is usually trained from large corpora such as newspapers. These corpora are too general in topic, so the resulting trigger pairs are not task-dependent.

We propose an adaptation paradigm in which the trigger pairs are extracted, and their probabilities are estimated from the initial speech recognition results. The initial transcription, although erroneous, contains many of the keywords whose dependencies we want to model. Therefore, it is a good source for deriving task-dependent trigger pairs, which we expect to have a significant effect on perplexity and speech recognition accuracy in a rescoring framework. To the best of our knowledge, this is the first work on constructing a trigger-based language model from the initial transcription.

This approach, however, poses two problems. The first one is that the size of the training data, that is, the size of the initial transcription, is much smaller than that of a large corpus, so it might be insufficient to extract enough trigger pairs and to reliably estimate their probabilities. The second problem is that, since the initial transcription contains errors, we may obtain erroneous triggers in addition to correct trigger pairs. These erroneous trigger pairs can have a harmful effect, increasing the probabilities of wrong words.

In order to cope with the first problem, instead of extracting the trigger pairs from

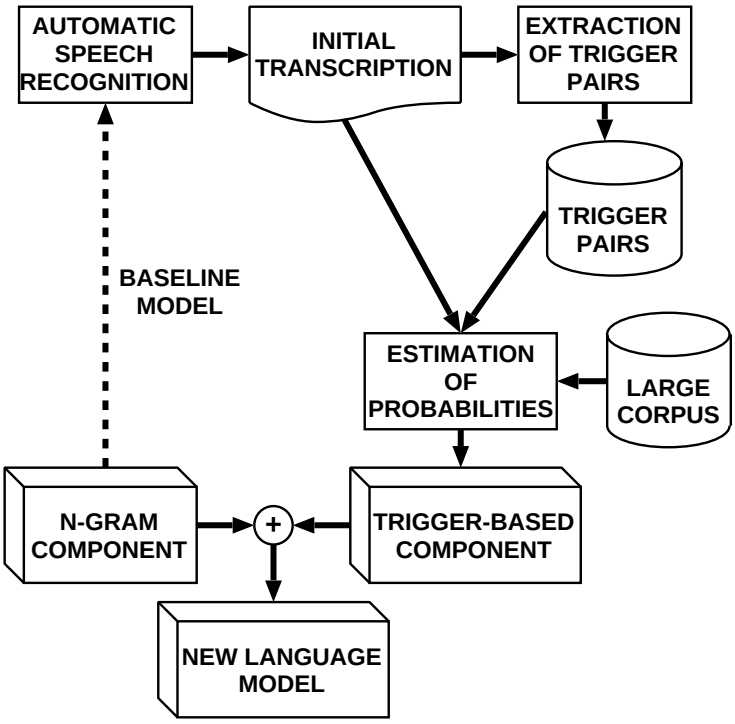

Figure 4.1: Outline of the proposed approach.

a window of fixed length with the average mutual information, we use the term frequency/inverse document frequency measure to find keywords from the whole document, and then we let any combination of two keywords be a candidate trigger pair. In this way, not only do we boost the possible number of trigger pairs, but we also capture topic constraints global to the document. In addition, since the probability estimates derived from the initial transcription might not be reliable, we propose a back-off scheme that incorporates statistics from a large corpus to the model.

As for the second problem, we use a confidence measure score to get rid of those trigger pairs whose component words are not reliable, while we assume that correct trigger pairs have a greater confidence score and consistently appear throughout the session. In this way we expect to minimize the number of incorrect trigger pairs.

Figure 4.1 illustrates the outline of the proposed approach. First, ASR is performed with a standard n-gram as the baseline language model, yielding the initial speech recognition results. The trigger pairs are then extracted and their probabilities are estimated from the initial transcription, as well as from a large corpus. Finally, the resulting trigger-based component is combined with the n-gram component to produce a new language model for the second pass of speech recognition.

4.3 Extraction of Trigger Pairs from Initial Transcription

This section details the extraction of trigger pairs from the initial speech recognition results.

4.3.1 Extraction based on TF/IDF instead of mutual information

Task-dependent trigger pairs are extracted from the initial transcription, namely the K-best ASR hypotheses. For the selection of pairs, instead of the average mutual information (AMI) used in [47, 56], we use the term frequency/inverse document frequency (TF/IDF) measure [59]. We employ this measure because it is document-based, that is, it lets us extract the trigger pairs from a whole document, rather than from a text window of the corpus. In this way, we can capture global constraints from each document. This measure is also chosen because of its simplicity.

The TF/IDF value of a term t_k in a document D_i is computed as follows:

$$v_{ik} = \frac{tf_{ik} \log\left(N/df_k\right)}{\sqrt{\sum_{j=1}^{T} (tf_{ij})^2 [\log(N/df_j)]^2}} \tag{4.1}$$

where tf_{ik} is the frequency of occurrence of t_k in D_i, N is the total number of documents, df_k is the number of documents that contain t_k, and T is the number of terms in D_i.

Since the initial transcription intuitively consists of only one document, the TF part (tf_{ik} and T) is calculated from the K-best hypotheses, whereas the IDF part (N and df_k) is computed from a fraction of a large corpus similar to the target task.

4.3.2 Part-of-speech and stop word filtering

We create all possible word pairs, including pairs of the same words (self-triggers), with the base forms and parts of speech (POS) of all content words with a TF/IDF value above a threshold. By regarding any combination of content words as a trigger pair, even though the size of the initial transcription is small, we obtain a large list of candidate trigger pairs. By using base forms we avoid same-root triggers, and we can apply the trigger pair when a word is presented with any inflection, while by using the POS information we distinguish between homonyms with different POS when applying the trigger pairs.

POS-based filtering is introduced to discard function words, and a stop word list with the most frequent words is used to ignore them during the extraction.

Table 4.3 shows some examples of trigger pairs extracted from the initial transcription of the target task that were actually used in the experiments. A bigger list can be found in appendix A.

4.3.3 Filtering with confidence score and large corpus

In order to minimize the adverse effect of erroneous trigger pairs, we introduce two methods to get rid of as many incorrect triggers as possible. First, we use the confidence score that is provided by the ASR system to eliminate the trigger pairs whose component words have a confidence score lower than a threshold.

Then, we compare the trigger pairs extracted from the initial transcription with pairs extracted from a large corpus, and we discard the trigger pairs which are not present in the second set.

With these methods, we can extract reliable trigger pairs, which are matched to the target domain.

Table 4.3: Example of trigger pairs extracted from the initial transcriptions of Sunday Discussion.

Triggering word	Triggered word
roudou (work)	*shifuto* (shift)
ame (rain)	*kasa* (umbrella)
shishutsu (expenses)	*kyasshu* (cash)
juutaku (housing)	*yachin* (rent)
isuramu (Islam)	*shuukyou* (religion)
mukashi (past)	*juurai* (former)
sodateru (to bring up)	*kyouiku* (education)
risuku (risk)	*kaihi* (avoidance)
teate (allowance)	*kyuufu* (payment)
kokusai (international)	*seiji* (politics)

4.4 Probability Estimation and Back-off Method

This section describes the probability estimation of the trigger pairs from the initial transcription, as well as a back-off scheme to incorporate trigger-based statistics derived from a large corpus.

4.4.1 Probability estimation from initial transcription

The probabilities of the trigger pairs are estimated from the K-best ASR hypotheses by using a text window to calculate the co-occurrence frequency of the pairs inside it. Given a trigger pair $w_1 \rightarrow w_2$, this text window consists of the L words preceding w_2.

The probability of each trigger pair is computed as follows:

$$P_{TP}^{IT}(w_2|w_1) = \frac{N(w_1, w_2)}{\sum_j N(w_1, w_j)} \tag{4.2}$$

where $N(w_1, w_2)$ denotes the number of times the words w_1 and w_2 co-occur within the text window, and j runs throughout all words triggered by w_1.

4.4.2 Proposed trigger-based language model

The proposed trigger-based language model is then constructed by linearly interpolating the probabilities of the trigger pairs with those of the baseline n-gram model, so that both long and short-distance dependencies can be captured at the same time.

The probability of the proposed language model for a word w_i given the word history $H = w_{i-L}, \cdots w_{i-1} \triangleq w_{i-L}^{i-1}$ is computed in the following way:

$$P_{LM}(w_i|H) = \frac{1}{L} \sum_{j=i-L}^{i-1} P_{LM}(w_i|w_j) \tag{4.3}$$

$$P_{LM}(w_i|w_j) = \begin{cases} P_{NG}(w_i|w_{i-n+1}^{i-1}), \text{ if } P_{TP}^{IT}(w_k|w_j) = 0, \forall k \\ \lambda P_{NG}(w_i|w_{i-n+1}^{i-1}) + (1-\lambda) P_{TP}^{IT}(w_i|w_j), \text{ otherwise} \end{cases} \tag{4.4}$$

Here L is the number of words in the history H; P_{NG} is the probability of the n-gram component, which uses only the last $n-1$ words of H (i.e. $n \ll L$); P_{TP}^{IT} is the probability of the trigger-based component, computed by equation (2); and λ is the language model interpolation weight. When there are no words triggered by w_j, the n-gram model alone is used. Otherwise, the n-gram probabilities are linearly interpolated with the probabilities from the trigger pairs.

4.4.3 Back-off method using statistics from large corpus

Since the amount of data provided by the initial transcription may be insufficient to obtain reliable probability estimates, a back-off scheme is introduced to combine the proposed model with the statistics estimated from a large corpus.

Another set of trigger pairs is extracted with the TF/IDF measure from a fraction of the large corpus similar to the target task. Then, the probabilities of the trigger pairs are computed from the whole corpus. We demonstrated in the previous chapter that the method that selects trigger pairs from a matched corpus and estimates their statistics with a larger corpus is effective. The resulting trigger pairs are similar to those used in the conventional trigger-based language model, except that the trigger pairs presented here are derived with the TF/IDF measure, instead of the AMI, and that they are extracted from a matched portion of the large corpus, instead of from the whole training set.

Then, we make use of this model to complement the proposed trigger-based language model described in section 4.4.2. We have two different sets of trigger pairs: the trigger pairs constructed from the initial transcription (hereafter trigger set IT), and the trigger pairs extracted from the large corpus (hereafter trigger set LC). The trigger set IT provides a probability distribution more faithful to the target domain, whereas the trigger set LC offers a more reliable distribution that can cope with the problem of data sparseness that we discussed in the previous chapter.

The probability of the enhanced language model based on the back-off scheme is calculated in the following way:

$$P_{BO}(w_i|w_j) = \begin{cases} P_{NG}(w_i|w_{i-n+1}^{i-1}), \text{ if } P_{TP}^{IT}(w_k|w_j) = 0, P_{TP}^{LC}(w_l|w_j) = 0, \forall k, l \\ \lambda P_{NG}(w_i|w_{i-n+1}^{i-1}) + (1-\lambda) P_{TP}^{LC}(w_i|w_j), \text{ if } P_{TP}^{IT}(w_k|w_j) = 0, \forall k \\ \lambda P_{NG}(w_i|w_{i-n+1}^{i-1}) + (1-\lambda)\big(\delta P_{TP}^{LC}(w_i|w_j) + (1-\delta) P_{TP}^{IT}(w_i|w_j)\big), \text{otherwise} \end{cases} \tag{4.5}$$

Here, P_{NG} is the probability of the n-gram component; P_{TP}^{IT} is the probability of the trigger set IT; P_{TP}^{LC} is the probability of the trigger set LC; λ is the language model interpolation weight; and δ is the trigger set interpolation weight. When there are no words triggered by w_j in either of the two trigger sets, the n-gram model alone is used. When there are no trigger pairs for w_j in the trigger set IT, the n-gram probabilities and the trigger set LC probabilities are linearly interpolated. Otherwise, all language models are linearly interpolated.

Note that if the trigger set IT is empty, that is, if we do not use the trigger pairs extracted from the initial transcription, the resulting model (first two entries in equation (4)) is similar to the conventional trigger-based language model, that is, a model whose trigger pairs are constructed from a large corpus. The differences are those we have just discussed. Hereafter we call this model the quasi-conventional trigger-based language model.

Table 4.4: Experimental setup.

Test set	10 programs (15K words each)
ASR system	Julius 3.5-rc2
Baseline language model	CSJ + National Diet trigrams
Acoustic model	Triphone HMM from CSJ
Vocabulary	30K words
OOV rate	1.56%
Baseline word accuracy	55.2%
Oracle word accuracy	76.5%
Baseline perplexity	150

4.5 Perplexity Evaluation

In this section we present the experimental evaluation of the proposed language model by test-set perplexity.

4.5.1 Experimental setup

The ASR system Julius 3.5-rc2 [39] was used for speech recognition. The baseline language model was a linear interpolation of word trigram models constructed from the Corpus of Spontaneous Japanese (CSJ) [49] (3.5M words) and the minutes of the National Diet of Japan (71M words) with an interpolation weight of 0.5. The size of the vocabulary was 30K words, and the out-of-vocabulary (OOV) rate was 1.56%. The acoustic model was a shared-state triphone HMM trained with the CSJ [38]. The average word recognition accuracy with this baseline model was 55.2%. We obtained this relatively low accuracy because the utterances are truly spontaneous and often uttered very fast.

The minutes of the National Diet from the year 2001 (17M words) were used for calculating the IDF part used in the trigger pair extraction of the set IT and also to extract the trigger pairs of the set LC.

The experimental setup is summarized in table 4.4.

4.5.2 Parameter optimization

The parameters of all models were optimized by 2-fold cross-validation. The test data were divided into two and the first 5 TV programs were used to empirically tune the parameters used in the evaluation of the other 5 programs and vice versa. The parameters were optimized by means of the perplexity.

The optimal language model interpolation weight λ was, for each half, 0.55 and 0.56 for the proposed trigger-based model (equation (3)), 0.66 and 0.67 for the quasi-conventional model (equation (4) without last entry), and 0.55 and 0.57 for the back-off method (equation (4)). The value of λ is larger for the quasi-conventional model than for the proposed models, because the trigger pairs are not task-dependent in the former model and, therefore, less beneficial in the interpolation.

The resulting optimal trigger set interpolation weight δ was 0.06 and 0.08, the word

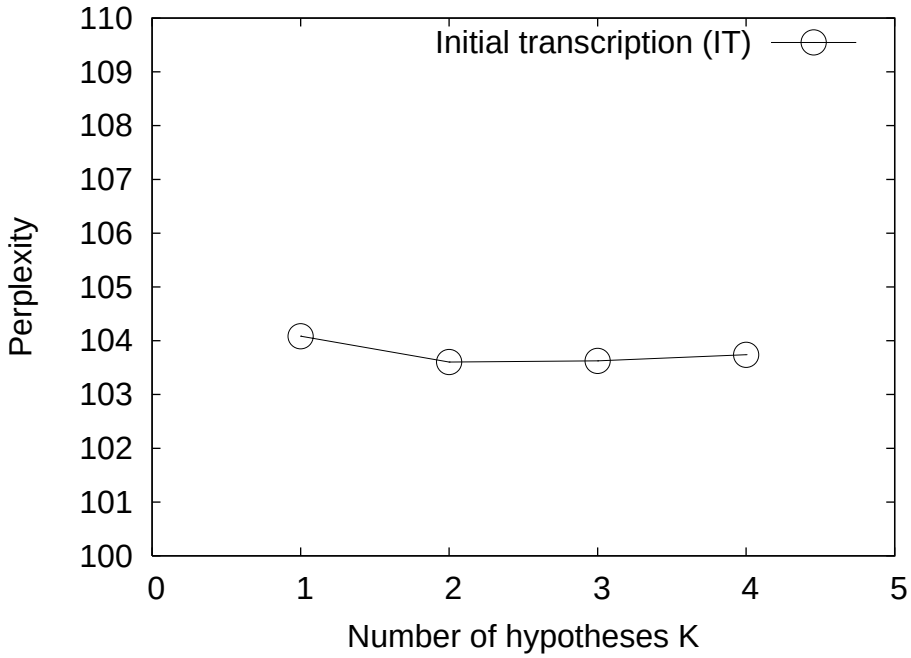

Figure 4.2: Perplexity of the proposed trigger-based language model for different values of the number of hypotheses K.

history size L was 25 and 28. The optimal number of hypotheses from the initial transcription K used for extracting the trigger pairs and estimating their likelihood was 2 for the two halves. Finally, the threshold for the TF/IDF value was 0.0005.

Figures 4.2, 4.3, 4.4, and 4.5 show the perplexity for different values of K, L, δ, and λ, respectively. We can see that the perplexity is not sensitive to the first three values, and that the perplexity changes smoothly with the language model interpolation weight λ.

Table 4.5 summarizes the results of parameter optimization.

In the experiments of perplexity evaluation, it turned out, after optimization, that the best performance was obtained when filtering with stop words, confidence score, and large corpus were not incorporated.

4.5.3 Experimental results

We evaluated the test-set perplexity for the 10 programs by three different models: the quasi-conventional trigger-based model using only a large corpus (LC), the proposed trigger-based language model using only the initial transcription (IT), and the back-off method (IT+LC). For reference, we also evaluated the model constructed by deriving the trigger pairs from the correct transcription.

The perplexity and its reduction averaged over the 10 programs are shown in Table 4.6. The proposed language model (IT) achieved a reduction of 30.66% over the baseline, much greater than the reduction obtained with the quasi-conventional model (LC). This demonstrates the effectiveness of the proposed approach.

The back-off method improved the perplexity slightly, but not significantly. This suggests that the initial transcription provides trigger pairs that are much more adapted

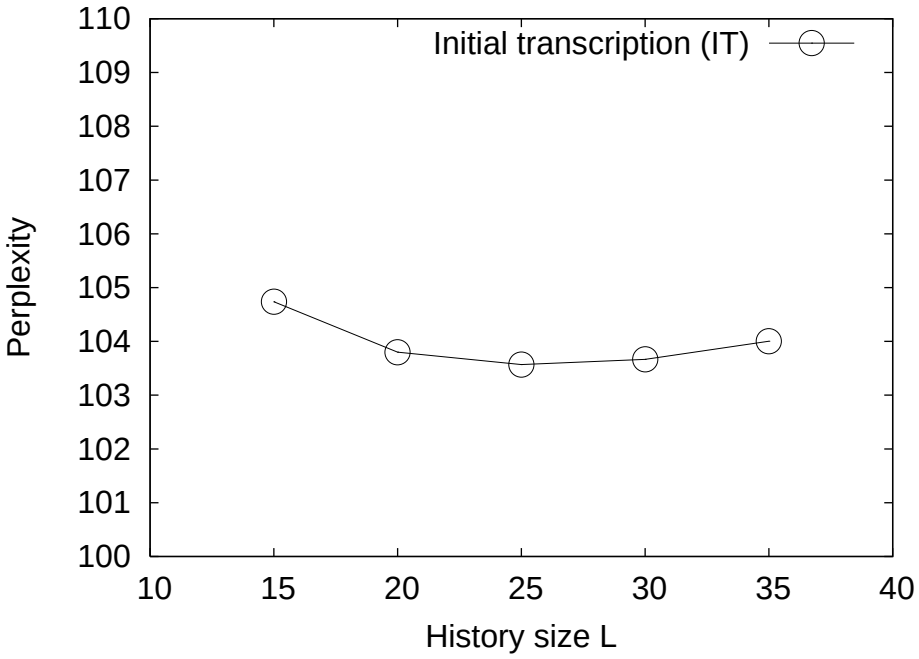

Figure 4.3: Perplexity of the proposed trigger-based language model for different values of the history size L.

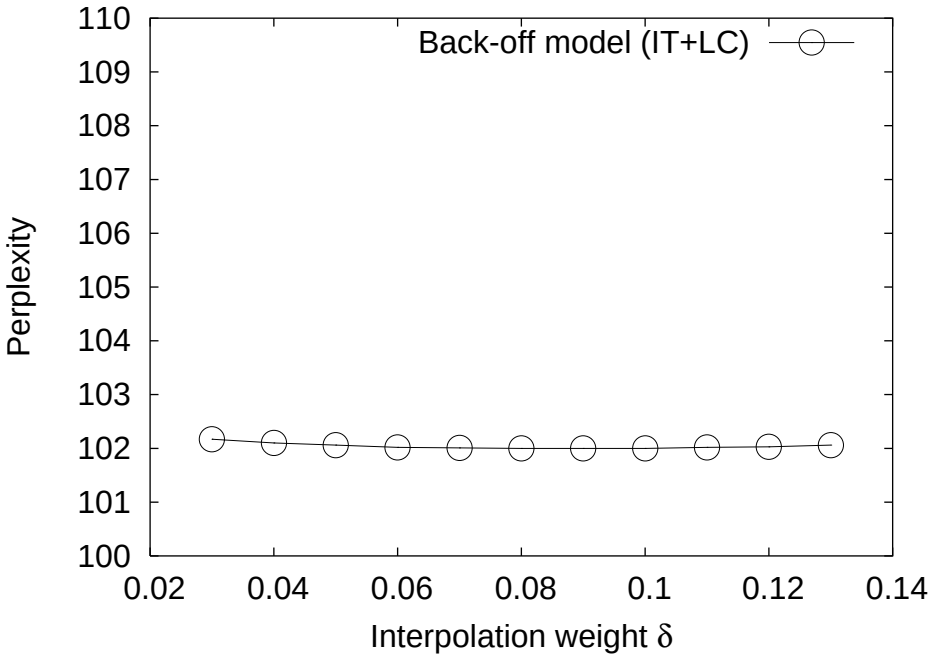

Figure 4.4: Perplexity of the proposed trigger-based language model for different values of the interpolation weight δ.

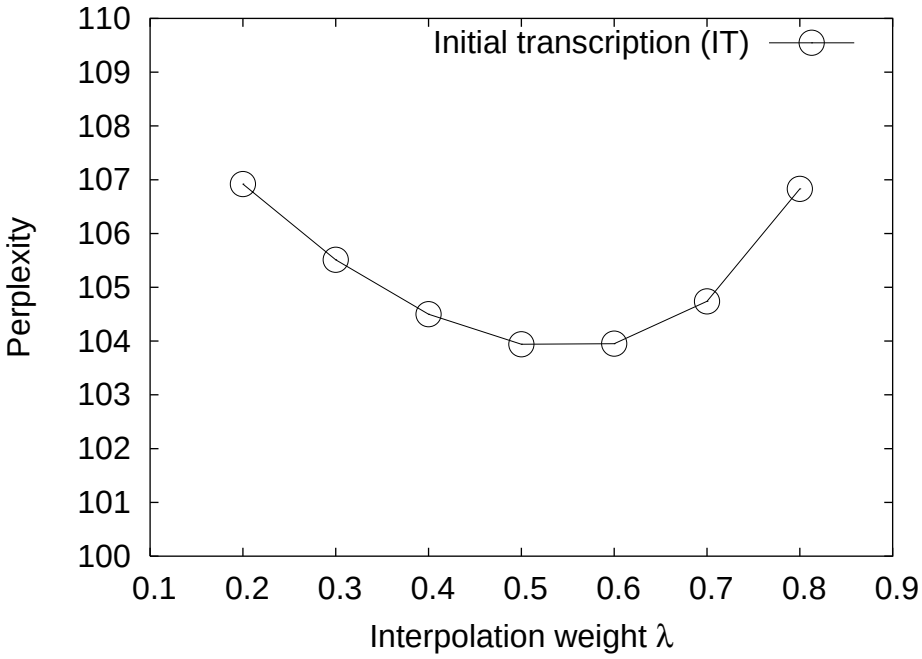

Figure 4.5: Perplexity of the proposed trigger-based language model for different values of the interpolation weight λ.

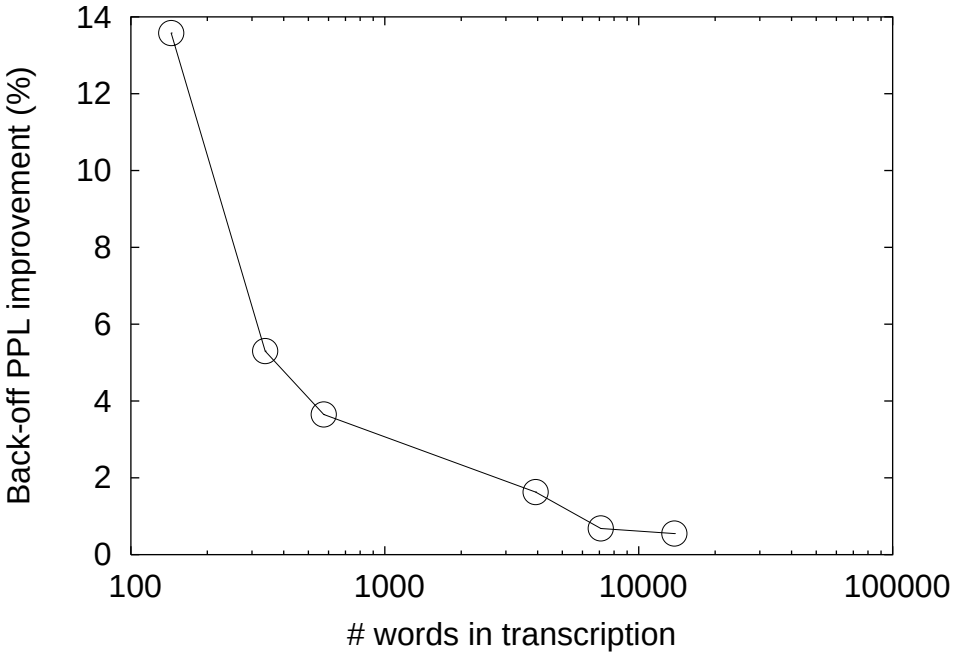

Figure 4.6: Perplexity improvement by the back-off model over the proposed trigger-based language model (IT) for different sizes of the initial transcription.

Table 4.5: Results of parameter optimization.

Parameter	Optimal value
Language model interpolation weight (λ)	0.66, 0.67 (LC); 0.55, 0.56 (IT); 0.55, 0.57 (IT+LC)
Trigger set interpolation weight (δ)	0.06, 0.08
Word history size (L)	25, 28
Number of hypotheses from IT (K)	2
Threshold for TF/IDF	0.0005

Table 4.6: Perplexity evaluation of trigger-based language models constructed by different methods.

Model	Perplexity	Reduction (%)
Baseline trigram	150	-
Large corpus (LC)	121	19.33
Initial transcription (IT)	104	30.66
Back-off model (IT+LC)	102	32.00
(cf.) Correct transcription	73	51.33

to the task than those constructed from the large corpus, so the benefit obtained from the latter is minimal. We expect that the proposed back-off scheme can be useful when the initial transcription is smaller in size. In order to support this claim, we calculated the perplexity improvement of the back-off model (IT+LC) over the proposed model constructed from the initial transcription (IT) for different sizes of the transcription. Figure 4.6 shows the results. We can see that the smaller the initial transcription, the better effect of the back-off method, as we expected.

The perplexity reduction by the proposed method was smaller than that obtained with the model that used the correct transcription. The baseline word recognition accuracy is 55.2%, meaning that about half of the initial transcription is erroneous, so the results are consistent with this fact.

We also constructed a trigger-based language model from the initial transcription by using the AMI [47, 56], instead of the TF/IDF measure. The perplexity was 104, which is comparable to that obtained when using the TF/IDF measure. It was observed that more trigger pairs were extracted by the TF/IDF measure, so we expect that this measure should be more effective for shorter discussions. In order to investigate this, we computed the perplexity improvement of the proposed method that uses the TF/IDF measure over the conventional method that uses the AMI for different sizes of the initial transcription. Figure 4.7 illustrates the results. We see that for smaller transcriptions the proposed method performs better than the conventional method based on the AMI.

Unlike the conventional works on the trigger-based language model, where trigger pairs are extracted by using a text window, the proposed approach extracts the trigger pairs from the whole discussion to capture global topic constraints. We compared these two approaches by creating trigger pairs by using a text window and comparing the

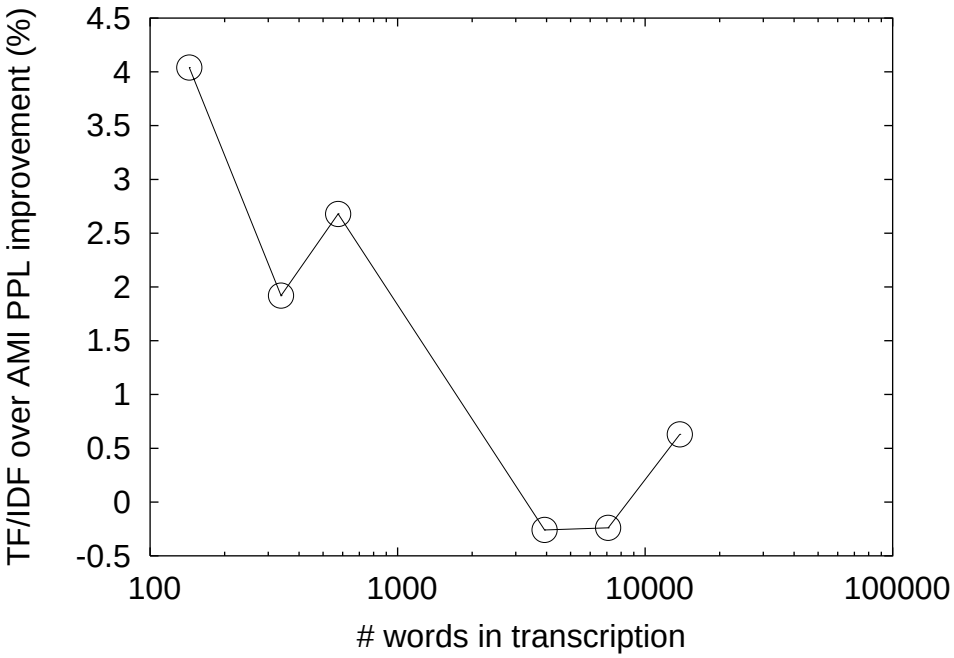

Figure 4.7: Perplexity improvement by the TF/IDF method over the AMI for different sizes of the initial transcription.

Table 4.7: Comparison of perplexity reductions for correctly recognized words and incorrectly recognized words.

Class of words	Model	Perplexity	Reduction (%)
Correctly recognized words	Baseline	75	-
	IT	49	34.66
Incorrectly recognized words	Baseline	408	-
	IT	298	26.96

perplexity reductions by this method and the proposed approach. The window size was optimized by the method explained in section 4.5.2, and the resulting optimal value was 8. The perplexity reduction by the trigger pairs constructed with the text window was 26%, lower than that obtained with the proposed approach. This proves that global topic constraints are more effective than local constraints in this task.

We also investigated the improvements for correctly recognized words and incorrectly recognized ones in the initial transcription. The average perplexity for correctly recognized words was 75 by the baseline model and 49 by the proposed model, whereas, for the incorrectly recognized words, the perplexity was 408 and 298, respectively. That is, we obtained a reduction of 34.66% for the correctly recognized words and an also significant 26.96% reduction for the incorrectly recognized ones. The fact verifies that the perplexity was also improved significantly for incorrect words, showing a potential of improvement in speech recognition accuracy. Table 4.7 illustrates this comparison.

The average number of trigger pairs was 128K in the trigger set IT, 9158K in the

Table 4.8: Number of used pairs and perplexity reductions when using only self-triggers and non-self-triggers from the initial transcription.

Model	Number of used pairs	Perplexity	Reduction (%)
Baseline trigram	-	150	-
Initial transcription (IT)	26K	104	30.66
Only self-triggers from IT	606	141	6.00
Only non-self-triggers from IT	26K	105	30.00

trigger set LC, and 71K from the correct transcription. The average hit rate of the trigger pairs in the test set was 31% for the first case, 33% for the second, and 35% for the third. We can see that the set IT efficiently covers the test set with a much smaller number of trigger pairs than the set LC. This is because the pairs from the set LC are not task-dependent. The back-off method had slight impact on the perplexity because the hit rate by using the set LC is only a little greater than that by the set IT.

The model constructed from the initial transcription (IT) used 606 self-triggers on average during the perplexity evaluation, while 26,555 non-self-triggers were used. The average perplexity when using only non-self-triggers was 105, very similar to that obtained when using all the trigger pairs, while the perplexity was 141 when using only self-triggers. Therefore, most of the perplexity reduction is due to non-self-triggers. This is a significant difference with the conventional works on trigger-based language models, where non-self-triggers offered little benefit over self-triggers. In contrast to previous works, the trigger pairs in the proposed approach are task-dependent and make a better match for the target task. We can see these results in table 4.8.

4.5.4 Comparison and combination with n-gram model adaptation

Next, we use the initial transcription also to create an adapted n-gram language model in order to compare its performance with that of the proposed approach. We then combine this with the proposed model for further improvement.

We take the J-best hypotheses from the initial transcription for creating a back-off n-gram model. A trigram model was constructed from each of the 10 test sets, and then interpolated with the baseline trigram model. The value J was optimized with the method discussed in section 4.5.2, yielding the value 10.

The resulting interpolated trigram was then combined with the trigger-based language model. Table 4.9 shows the results of the perplexity evaluation. The perplexity reduction by the n-gram adaptation is smaller than that by the proposed trigger-based adaptation, and their combination achieved a notable maximum perplexity reduction of 44% over the baseline trigram model. Although the improvement is not additive, the n-gram model adaptation serves as a good complement for the proposed approach.

Figures 4.8 and 4.9 show the perplexity by several of the constructed language models for each of the topics (test discussions) and speakers, respectively. As can be observed, the results are fairly consistent across the different topics and speakers.

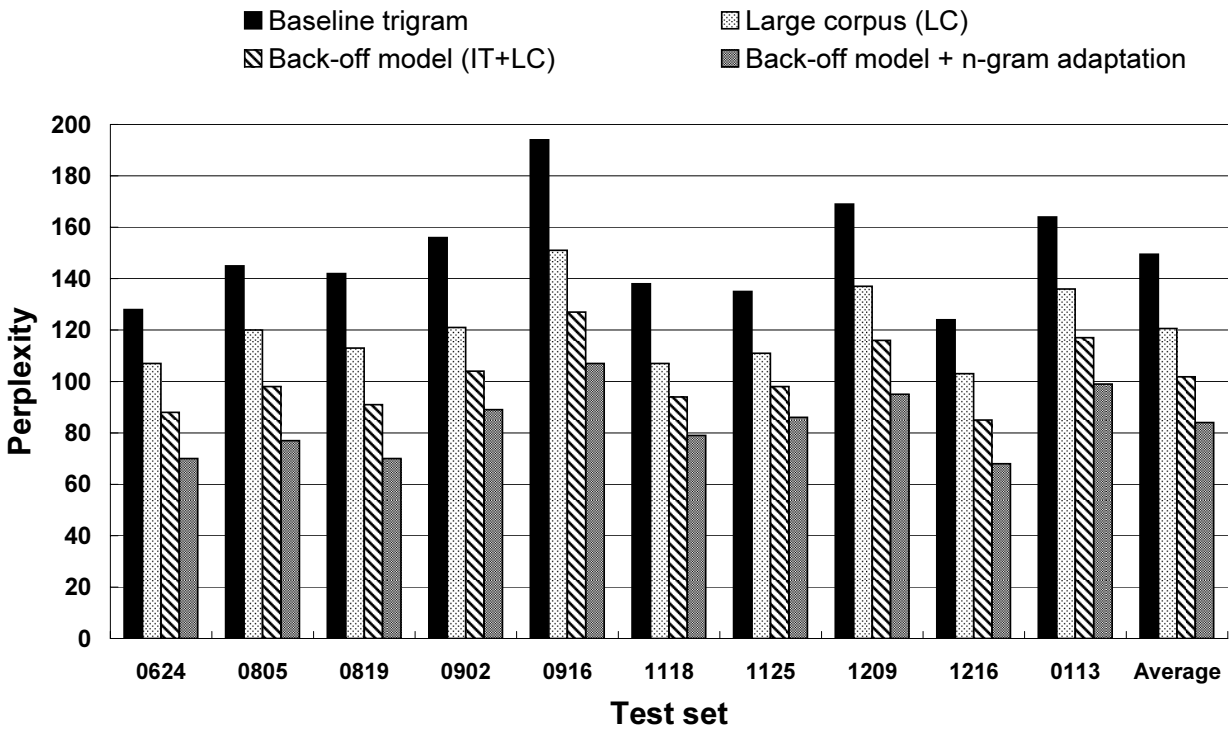

Figure 4.8: Perplexity evaluation of reference and proposed trigger-based language models among different topics.

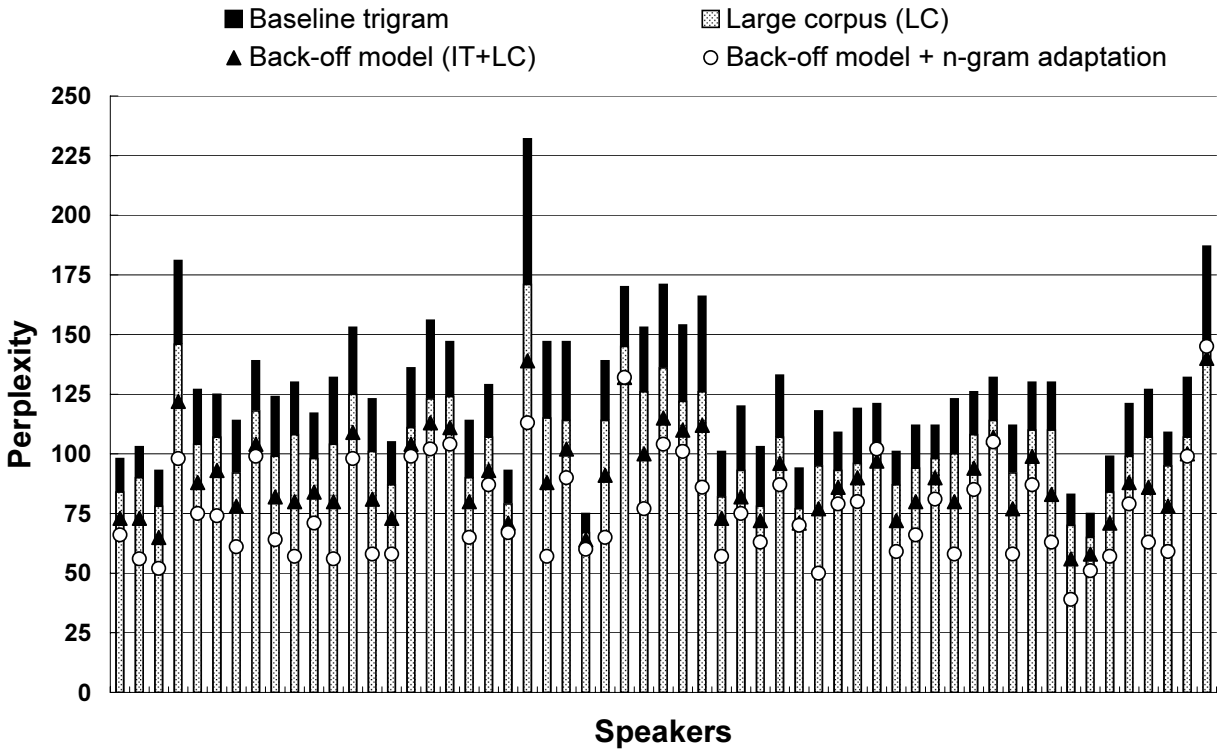

Figure 4.9: Perplexity evaluation of reference and proposed trigger-based language models among different speakers.

Table 4.9: Perplexity evaluation of the adapted n-gram and its combination with the proposed trigger-based language model.

Model	Perplexity	Reduction (%)
Baseline trigram	150	-
Adapted trigram	119	20.66
+Initial transcription (IT)	87	42.00
+Back-off model (IT+LC)	84	44.00

4.6 Speech Recognition Evaluation

This section presents a scheme for rescoring word graphs by the proposed language model and the experimental results in terms of speech recognition accuracy.

4.6.1 Word graph rescoring

The ASR system Julius generates a word graph with acoustic, language, and confidence scores for each node. The experimental setup is the same as in section 4.5.1.

Then, we use a stack decoding search for parsing the word graph to find the most likely sentence hypothesis [36]. During the search, we use the proposed trigger-based language model to recalculate the language model scores, by discounting the baseline language model probability from the per-node combined score and then adding the proposed language model probability. The word history is formed with the 1-best hypotheses of the preceding utterances and with the words that make up the partial path in the current utterance.

4.6.2 Experimental results

We evaluated the word error rate (WER) for each of the 10 programs of the test set. In this section, filtering with stop words, confidence score, and large corpus were incorporated. Here also, we conducted the two-fold cross validation described in section 4.5.2. The resulting confidence threshold was, for each half of the test set, 0.04 and 0.06, the frequency threshold for the stop word list was 200 for the two halves, and the average word history size was changed to 44 and 42.

Figure 4.10 shows the results obtained by the adapted trigram model, the proposed language model (IT), and those by the model constructed from the correct transcription. We obtained a relative 0.98% improvement in WER for the proposed language model. This improvement, although small, is statistically significant, with a p-value of 0.022. The adapted trigram achieved a relative 0.43% improvement, also comparatively smaller than its perplexity reduction.

We also examined the WER when using the AMI instead of the TF/IDF measure, and we obtained no significant difference.

In addition, we investigated the WER when the confidence score filtering and the large corpus filtering were alternatively used. When only the large corpus filtering was used, we obtained a 0.91% improvement over the baseline, while when only the confidence score

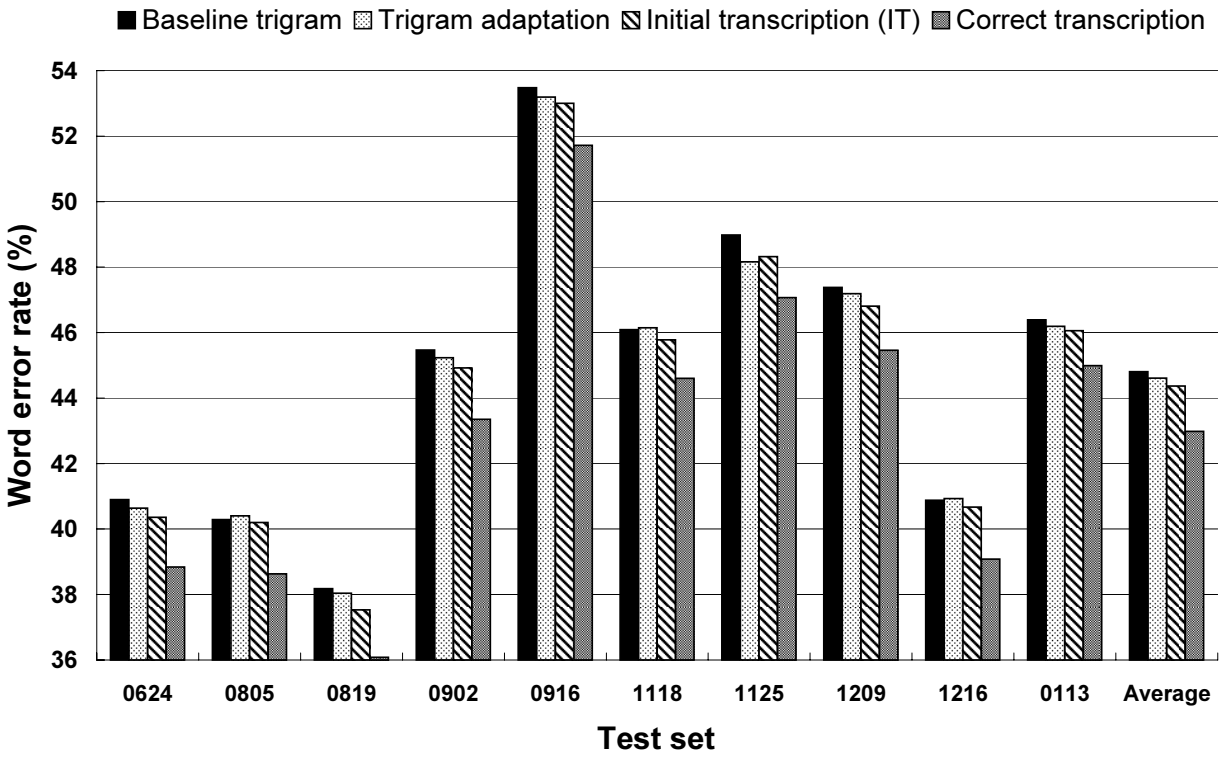

Figure 4.10: Word error rate improvement by the proposed trigger-based language model.

filtering was used, the improvement was 0.9%, as compared with the 0.98% improvement when using both filtering methods. Therefore, the two filtering methods have some effect in reducing erroneous trigger pairs. We can see this in table 4.10. Here, we compared the trigger pairs that were actually used during the rescoring experiments with those extracted from the correct transcription. In order to decide if the used trigger pairs were correct or incorrect, trigger pairs that did not appear in the list of pairs extracted from the correct transcription were labeled as incorrect. We can see in the table that the proposed filtering methods helped reduce the proportion of incorrect trigger pair entries from above 71% to around 56%, although some correct trigger pairs were also discarded.

The reasons why the obtained improvement in WER is much smaller than the perplexity reduction by the proposed language model are presumed as follows. First, as shown in figure 4.11, the proportion of incorrect trigger pairs used during the rescoring experiments (around half of them) is much greater than that during the perplexity experiments (less than 2%), where incorrect trigger pairs have little effect. Figure 4.11 compares the WER obtained by the proposed model with that obtained by the model that only uses correct triggers, whose relative WER reduction was 2.6%. Second, although the reduction in perplexity for incorrectly recognized words is significant, the perplexity value is still very large (reduced from 408 to 298), so it is hard to improve the recognition accuracy. Third, when we calculate the perplexity, the word history does not contain any errors, so the predictors are much better than those used in the speech recognition experiments. Conversely, the word history contains errors during the word graph rescoring, thus a history size greater than that used in the perplexity evaluation was needed. Finally, the word graph we rescore has the apparent limitation that the correct words might not be in any of the nodes. We expect that a re-decoding scheme with the adapted model would

Table 4.10: Distribution of correct and incorrect trigger pairs used during the rescoring experiments when confidence score filtering and large corpus filtering were used and not.

Confidence score	Large corpus filtering	Class of triggers	Entries	Count	Proportion (%)	
0	No	Correct	9504	35582	28.29	36.36
		Incorrect	24086	62264	71.71	63.64
0.04 / 0.06	No	Correct	8915	35152	35.42	44.44
		Incorrect	16251	43940	64.58	55.56
0	Yes	Correct	7870	31757	38.66	47.36
		Incorrect	12487	35297	61.34	52.64
0.04 / 0.06	Yes	Correct	7441	30290	43.91	52.88
		Incorrect	9505	26987	56.09	47.12

Table 4.11: Distribution of the total number of extracted correct and incorrect trigger pairs and of those used during the perplexity and speech recognition experiments.

	Class of triggers	Entries	Count	Proportion (%)	
Total pairs	Correct	31253	-	24.23	-
	Incorrect	97727	-	75.77	-
Pairs used in PPL experiments	Correct	14848	26716	97.37	98.36
	Incorrect	401	446	2.63	1.64
Pairs used in WER experiments	Correct	7441	30290	43.91	52.88
	Incorrect	9505	26987	56.09	47.12

realize a greater improvement as shown in [1, 50], whose perplexity reductions are much smaller than the one obtained in this work, and where n-gram adaptation had a significant improvement in re-decoding. With the correct transcription, the relative WER improvement was 4.07%, much greater than that obtained with the initial transcription, so we anticipate better results in tasks with higher baseline ASR performance.

4.7 Application to the National Diet Corpus

In this section we apply the proposed approach to the National Diet corpus, and present perplexity and speech recognition evaluation results.

4.7.1 Task and procedure

The target task here is the transcription of the National Diet (Congress) of Japan [1]. One session divided into three chunks of two hours were used as the test data, totaling 63929 words, with an average number of utterances and words per data set equal to 100 and 21K, respectively.

Topics in the National Diet change abruptly during the sessions, so instead of extract-

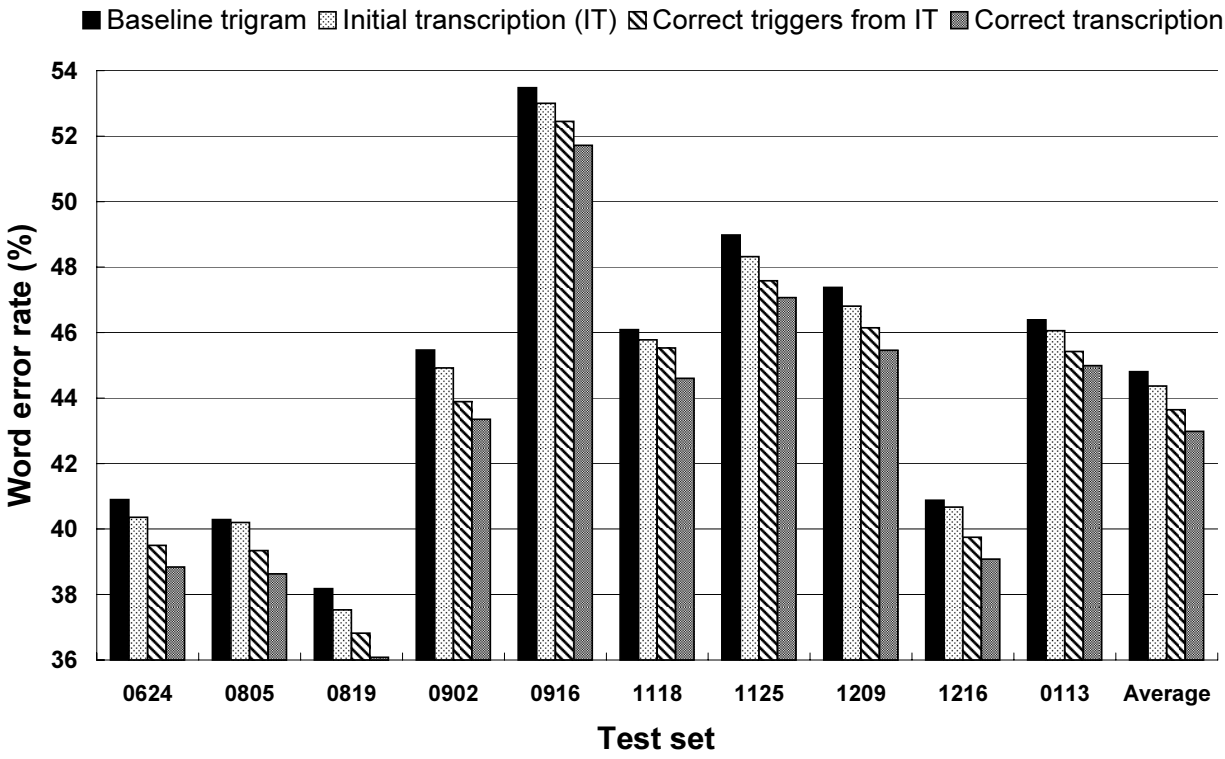

Figure 4.11: Word error rate improvement by the trigger-based language model that uses only correct trigger pairs.

ing the trigger pairs from the whole session as we did in the panel discussions task, here trigger pairs are extracted by using a text window. In this way we can capture local topic constraints and construct a model more robust to sudden topic shifts. Apart from this change, the trigger-based language model was constructed as in the previous task.

Table 4.12 shows some examples of trigger pairs extracted from the initial transcription of the target task that were actually used in the experiments. A bigger list can be found in appendix A.

4.7.2 Perplexity evaluation

The experimental setup is similar to that used in the previous task, and it is summarized in table 4.13. This time, the baseline word recognition accuracy is 68.5%, higher than the 55.2% obtained for the panel discussions task, and the perplexity is 125, lower than the value 150 obtained for the previous task. Since the initial transcription this time has less errors than that of the previous task, and the perplexity is lower, we expect to achieve better results both in terms of perplexity and speech recognition accuracy.

The parameters of all models were optimized by leave-one-out cross-validation. One of the data sets was used as the test data and the other two were used to empirically tune the parameters of the models. This was repeated until all the three data sets were used as the test data.

The optimal language model interpolation weight λ was 0.6 for the proposed trigger-based model (equation (3)), 0.66 for the quasi-conventional model (equation (4) without last entry), and 0.55 for the back-off method (equation (4)). The resulting optimal trigger set interpolation weight δ was 0.1, the word history size L was 20. The optimal number

Table 4.12: Example of trigger pairs extracted from the initial transcription of the National Diet.

Triggering word	Triggered word
keikaku (plan)	*kaihatsu* (development)
iraku (Iraq)	*heiki* (weapon)
rachi (abduction)	*kitachousen* (North Korea)
heiki (weapon)	*sensou* (war)
nenkin (pension)	*okane* (money)
toushi (investment)	*chochiku* (savings)
sekiyu (petroleum)	*enerugii* (energy)
shigen (resource)	*kankyou* (environment)
shiberia (Siberia)	*roshia* (Russia)
gasu (gas)	*saharin* (Sakhalin)

Table 4.13: Experimental setup.

Test set	One session divided into 3 data sets (21K words each)
ASR system	Julius 3.5-rc2
Baseline language model	CSJ + National Diet trigrams
Acoustic model	Triphone HMM from CSJ
Vocabulary	30K words
OOV rate	1.45%
Baseline word accuracy	68.5%
Baseline perplexity	125

of hypotheses from the initial transcription K used for extracting the trigger pairs and estimating their likelihood was 3. Finally, the threshold for the TF/IDF value was 0.0005. Table 4.14 summarizes the results of parameter optimization.

In the experiments of perplexity evaluation, it turned out, after optimization, that the best performance was obtained when stop word list, confidence score, and large corpus filtering were not incorporated.

We evaluated the test-set perplexity for the three data sets by three different models: the quasi-conventional trigger-based model using only a large corpus (LC), the proposed trigger-based language model using only the initial transcription (IT), and the back-off method (IT+LC). For reference, we also evaluated the model constructed by deriving the trigger pairs from the correct transcription.

The perplexity and its reduction averaged over the three data sets are shown in Table 4.15. These results are similar to those obtained for the Sunday discussion task. The proposed language model (IT) achieved a reduction of 32.80% over the baseline, much greater than the reduction obtained with the quasi-conventional model (LC). This demonstrates the effectiveness of the proposed approach. As in the previous task, the back-off scheme improved the perplexity slightly.

Figure 4.12 shows the perplexity by several of the constructed language models for

Table 4.14: Results of parameter optimization.

Parameter	Optimal value
Language model interpolation weight (λ)	0.66 (LC); 0.6 (IT); 0.55 (IT+LC)
Trigger set interpolation weight (δ)	0.1
Word history size (L)	20
Number of hypotheses from IT (K)	3
Threshold for TF/IDF	0.0005
Window size	8

Table 4.15: Perplexity evaluation of trigger-based language models constructed by different methods.

Model	Perplexity	Reduction (%)
Baseline trigram	125	-
Large corpus (LC)	101	19.20
Initial transcription (IT)	84	32.80
Back-off model (IT+LC)	83	33.60
(cf.) Correct transcription	60	52.00

each of the data sets of the National Diet. As can be observed, the results are fairly consistent across the different test data.

We also investigated the perplexity improvements for correctly recognized words and incorrectly recognized ones. The average perplexity for correctly recognized words was 96 by the baseline model and 64 by the proposed model, whereas, for the incorrectly recognized words, the perplexity was 273 and 184, respectively. That is, we obtained a reduction of 33.33% for the correctly recognized words and a 32.60% reduction for the incorrectly recognized ones. Table 4.16 illustrates this comparison. In this case, the perplexity reduction for incorrectly recognized words was very similar to that for correctly recognized words, and better than in the previous task. Since we are using longer initial transcriptions than in the previous task (21K words vs. 14K words), we end up with better probability estimates for the trigger pairs, thus the greater reduction for incorrectly recognized words. As a matter of fact, the average trigger probability for this task was 0.037, while it was 0.013 for the previous task.

4.7.3 Speech recognition evaluation

We evaluated the WER for each of the three test data sets. In this section, filtering with confidence score and large corpus were incorporated. Here also, we conducted the leave-one-out cross-validation described in the previous subsection. The resulting average confidence threshold was 0.15, and the average word history size was changed to 40.

Figure 4.13 shows the results obtained by the proposed language model (IT) and those by the model constructed from the correct transcription. We obtained a relative 1.20% improvement in WER for the former model and a relative 4.20% improvement

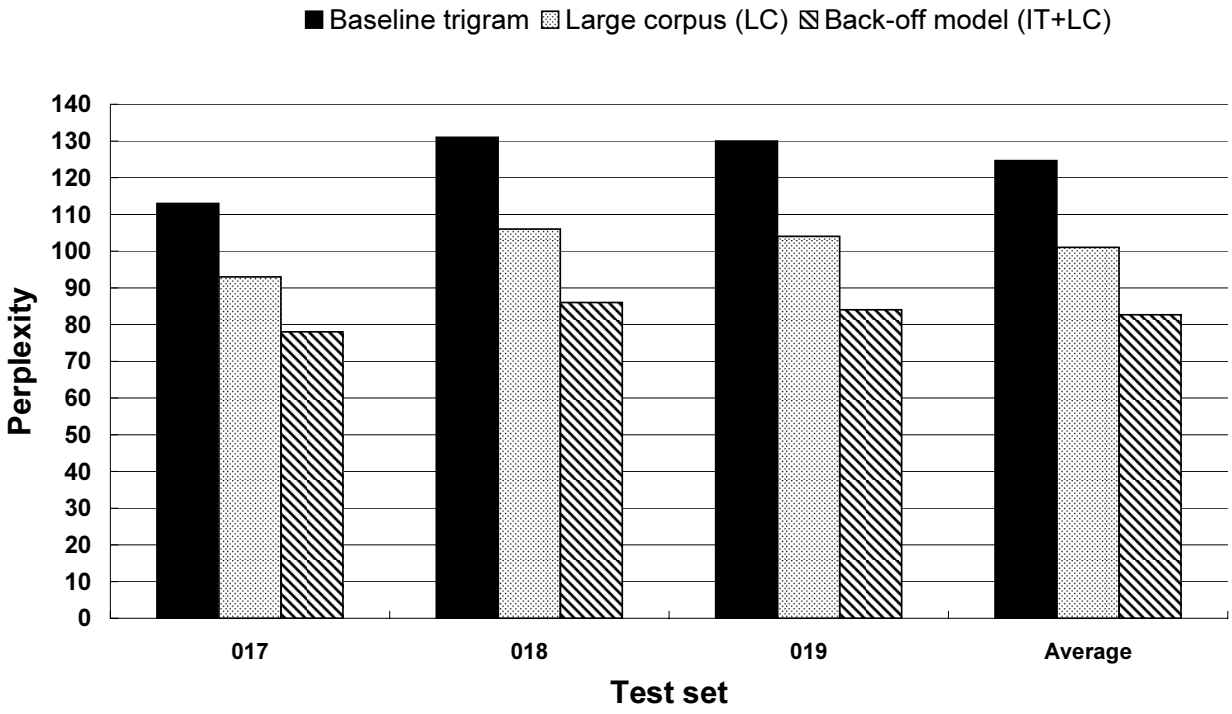

Figure 4.12: Perplexity evaluation of reference and proposed trigger-based language models for different data sets.

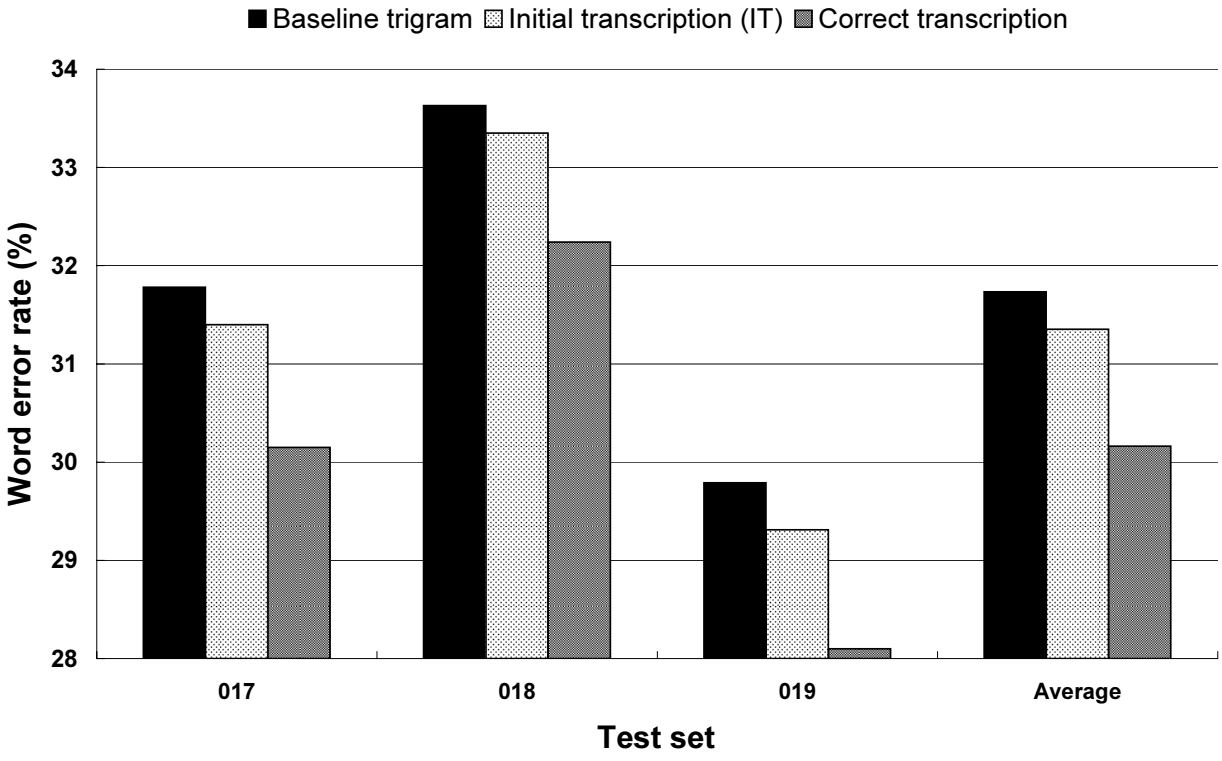

Figure 4.13: Word error rate improvement by the proposed trigger-based language model for the National Diet task.

Table 4.16: Comparison of perplexity reductions for correctly recognized words and incorrectly recognized words.

Class of words	Model	Perplexity	Reduction (%)
Correctly recognized words	Baseline	96	-
	IT	64	33.33
Incorrectly recognized words	Baseline	273	-
	IT	184	32.60

Table 4.17: Distribution of the total number of extracted correct and incorrect trigger pairs and of those used during the rescoring experiments.

	Class of triggers	Entries	Count	Proportion (%)	
Total pairs	Correct	18120	-	24.47	-
	Incorrect	55932	-	75.53	-
Used pairs	Correct	8776	158299	58.20	64.50
	Incorrect	6363	88974	41.80	35.50

for the latter. These improvements are greater than the 0.98% and 4.07% respective improvements obtained in the previous task. As we mentioned, the higher word accuracy in this task makes the initial transcription a less erroneous source for extracting the trigger pairs, thus the smaller number of erroneous trigger pairs is less harmful.

We compared the total number of extracted trigger pairs, and those that were actually used during the rescoring experiments with the proposed language model (IT). Table 4.17 shows the results. We can see that the proportion of incorrect trigger pairs is lower in this task (41.80%) than in the previous one (56.09%), since the baseline word accuracy is higher than in the Sunday Discussion task, thus the better improvement in performance.

4.8 Conclusion

We presented a novel trigger-based language model adaptation scheme, oriented to the transcription of meetings, based on initial speech recognition results. In meetings, the topic is focused and consistent throughout the whole session, therefore keywords can be correlated over long distances. The trigger-based language model is designed to capture such long-distance dependencies, but it is typically constructed from a large corpus, which is usually too general to derive task-dependent trigger pairs. In the proposed method, we make use of the initial speech recognition results to extract task-dependent trigger pairs and to estimate their statistics. Moreover, we introduce a back-off scheme that also exploits the statistics estimated from a large corpus.

The proposed model reduced the test-set perplexity considerably more than the typical trigger-based language model constructed from a large corpus, and achieved a remarkable perplexity reduction of 44% over the baseline when combined with an adapted trigram language model. Furthermore, it was observed that, contrary to the common finding in

conventional works on the trigger-based language model, much more non-self-triggers were used in the proposed method. This demonstrates that the proposed approach, as opposed to the typical trigger-based language model, effectively constructs task-dependent trigger pairs from the available in-domain data. In addition, a reduction in word error rate was obtained when using the proposed language model to rescore word graphs.

The proposed approach is particularly useful in tasks where large amounts of training data are not readily available but the test set is sufficiently long, since we have observed that the initial transcription is a good source for deriving the trigger pairs. This is specifically true for many transcription tasks. A further study of the applicability of this approach is presented in the next chapter.

Chapter 5

Conclusion

5.1 Summary and contributions

5.1.1 Summary

This thesis presented two different approaches of trigger-based language modeling for the transcription of conversational speech. Both approaches take advantage of the available in-domain data to derive task-dependent trigger pairs, while they make use of a large corpus to reliably estimate their statistics.

The first approach was presented in chapter 3, where the trigger pairs were extracted from the task corpus and their probabilities were estimated from both the task corpus and the large corpus. This trigger-based language model was applied to two different conversational speech tasks, and it achieved significant improvements in test-set perplexity and also improved the word recognition accuracy with N-best rescoring.

The second approach, presented in chapter 4, is trigger-based language model adaptation for smaller amounts of in-domain data. In this case, the trigger pairs were extracted from the initial speech recognition results, and their probabilities were also estimated from this information source. A back-off scheme was then used to combine the statistics of the trigger pairs constructed from the initial transcription with those constructed from a large corpus. This method was used for two different transcription tasks, achieving a remarkable perplexity reduction and also a significant reduction in WER when rescoring word graphs.

5.1.2 Contributions

The trigger-based language model has been mainly applied to the recognition of newspaper tasks, and it has been typically constructed from large corpora such as newspapers [47, 56, 74, 5]. In this thesis, instead of written-style tasks, we applied the trigger-based language model to the transcription of conversational speech.

Large corpora are usually too general in topic and do not closely match the specific test data, thus the trigger pairs constructed from them are not task-dependent. In this research, task-dependent trigger pairs that closely match the target task were extracted from the available in-domain data. In addition, since the probability estimates derived from the target domain might not be reliable, because of the typical small amount of data, we proposed a back-off scheme that incorporates the statistics from the large corpus

to the model. Moreover, the trigger pairs are usually constructed from a text window of fixed length with the average mutual information measure. This window limits the scope of the dependencies the trigger-based language model can capture. We used the TF/IDF measure to extract the trigger pairs from the whole document, instead of a text window, to capture topic constraints global to the document.

A common finding in trigger-based language modeling is that much of the potential of these models lies in words that trigger themselves, called self-triggers, which are virtually equivalent to the cache-based language model, so the original trigger-based language model does not significantly outperform the cache-based model. During their evaluation, the proposed trigger-based language models used much more non-self-triggers than self-triggers, and most of the perplexity reduction was due to non-self-triggers, which is a significant difference with the conventional trigger-based language model. This is because the trigger pairs in the proposed approach are task-dependent and make a better match for the target task.

To the best of our knowledge, this is the first work that constructs a trigger-based language model from the initial speech recognition results. Finally, the literature on trigger-based language models applied to Japanese corpora is almost inexistent, so this is another contribution of the present research, where the trigger-based model was applied to four different Japanese tasks.

5.1.3 Applicability

The proposed trigger-based language models are intended for tasks where large amounts of training data are not readily available, since we proved that large corpora can complement the available in-domain data with the proposed methods. This is specifically true for spoken language tasks, where available corpora are typically small.

In addition, in order to get the most from the proposed research, it should be applied to tasks that can be divided into documents based on topics, and where the topics are well defined. The more homogeneous the topic, the more topic-dependent trigger pairs can be extracted.

The proposed adaptation based on initial transcriptions should be used in tasks with higher baseline word recognition accuracy. As the recognition accuracy is improved, less erroneous trigger pairs are extracted, so the harmful effect of these is reduced. We proved that the back-off scheme should be advantageous for transcriptions shorter than the ones used in this work, since in that case we expect the statistics from the large corpus to account for the data sparseness problem.

We conclude that broadcast news should be an appropriate task for applying this approach, since topics in broadcast news are explicit, because each news story focuses on a given subject matter, typical broadcast news tasks have a high speech recognition accuracy, and news stories are typically short.

5.2 Future directions

Among the directions we consider worth exploring in the future are the following. First, it would be beneficial to use an exponential model such as the maximum entropy framework or log-linear interpolation to combine models, instead of the suboptimal linear interpolation scheme [6].

Second, instead of the whole large corpus, we could use only the text from it that is similar to the target task to use only relevant data for the back-off scheme. We propose here two different ways to do this. One possible way is to find in the large corpus documents similar to the target task, for example by using probabilistic latent semantic indexing (PLSI) [24], and using only the data from those documents. The other one is, instead of using the document as the unit, to choose only the sentences from the large corpus that are similar to those of the target task, by using some similarity measure such as BLEU, similarly to the method proposed in [60].

Finally, it would be optimal to use re-decoding instead of rescoring for integrating the language model with the speech recognizer, and we also encourage the application of the proposed trigger-based language to broadcast news corpora.

Appendix A

Lists of Trigger Pairs

In this appendix we show lists of trigger pairs created from each of the corpora used in this thesis. For each corpus, the best 100 trigger pairs (excluding self-triggers) ranked by their frequency in the corpus are shown, from the most frequent to the least frequent of the 100. The frequencies were calculated by using the same text window that was used during the probability estimation of the trigger pairs.

This appendix is divided into two sections. The first one lists trigger pairs extracted from the training data of each corpus, with the method explained in chapter 3, while the second one enumerates trigger pairs extracted from the initial transcription of each task, as described in chapter 4.

A.1 Trigger pairs extracted from training data

This section presents lists of trigger pairs extracted from the training data of all the corpora used in chapter 3.

A.1.1 Trigger pairs from Mainichi Shimbun

容疑 → 逮捕
記者 → 会見
政治 → 改革
金融 → 機関
逮捕 → 容疑
事件 → 容疑
死去 → 葬儀
黒 → 白
白 → 黒
官房 → 長官
葬儀 → 告別
午前 → 午後
事件 → 捜査
葬儀 → 喪主
容疑 → 調べ
自宅 → 喪主
捜査 → 容疑
衆院 → 議員
死去 → 告別
大統領 → 会談
死去 → 喪主
葬儀 → 自宅
クリントン → 大統領
国会 → 議員
午後 → 自宅
告別 → 喪主
死去 → 自宅
不良 → 債権
午後 → 喪主
ロシア → 大統領
選挙 → 候補
首脳 → 会談
選挙 → 党
告別 → 自宅
事件 → 被告
病院 → 死去
核 → 実験
自民党 → 党
病院 → 葬儀
午前 → 自宅
政治 → 選挙
党 → 選挙
捜査 → 本部
選挙 → 政治
病院 → 自宅
エリツィン → 大統領
阪神 → 大震災
午後 → 午前
調べ → 容疑
病院 → 告別
会社 → 社長
午前 → 喪主
葬儀 → 午後
議員 → 選挙
改革 → 政治
調査 → 結果
事件 → 逮捕
疑い → 容疑
自民党 → 政治
昨年 → 今年
大会 → 優勝
日 → 関係
選挙 → 投票
自民党 → 選挙
病院 → 喪主
自民党 → 議員
会議 → 開く
容疑 → 疑い
容疑 → 事件
告別 → 午後
男性 → 女性
大統領 → 選挙
午後 → 葬儀
世界 → 選手権
関係 → よる
選挙 → 自民党
死去 → 午後
開く → 会議
首脳 → 会議
容疑 → 捜査
候補 → 選挙
証券 → 取引
会社 → 容疑
参院 → 議員
会談 → 大統領
携帯 → 電話
大阪 → 容疑
経済 → 改革
昨年 → 十二月
選挙 → 議員
大統領 → ロシア
電話 → 番号
安全 → 保障
ブッシュ → 大統領
午後 → 病院
大会 → 出場
女性 → 男性
党 → 自民党
連立 → 政権
中小 → 企業

A.1.2 Trigger pairs from BTEC

予約 → 荷物
席 → 予約
予約 → 席
部屋 → 予約
時間 → 何時
予約 → 部屋
部屋 → いかが
席 → 何時
やる → 予約
いつ → 席
いかが → わかる
来る → 席
予約 → 教える
予約 → 何時
予約 → バス
部屋 → わかる
席 → よろしい
席 → いかが
警察 → 呼ぶ
荷物 → 席
営業 → 何時
ホテル → よい
バス → 予約
予約 → 待つ
予約 → 借りる
予約 → 時間
予約 → よろしい
予約 → ない
部屋 → 席
病院 → 連れる
入る → 事故
日本語 → わかる
探す → 予約
席 → 今日
席 → 教える
席 → 荷物
写真 → 撮る
見せる → 席
近く → 部屋
教える → 料金
教える → 現金
何時 → わかる
何時 → よろしい
ホテル → いう
ドル → 替える
わかる → 部屋
もっと → 予約
もっと → 安い
お金 → 部屋
いかが → 部屋
いう → よい
列車 → 席
料金 → 予約
料金 → 見せる
旅行 → 小切手
預かる → 予約
予約 → 料金
予約 → 必要
予約 → 入る
予約 → 二つ
予約 → 搭乗
予約 → 全部
予約 → 新しい
予約 → 見る
予約 → いかが
部屋 → 忘れる
入れる → 何時
入る → 予約
東京 → 予約
滞在 → 予定
待つ → 席
待つ → 今
待つ → 荷物
席 → 料金
席 → 入る
席 → 見る
席 → 見せる
席 → 一つ
席 → やる
場所 → 教える
出る → 料金
持つ → 何時
今日 → 予約
今日 → いう
今 → 予定
今 → 滞在
今 → 席
今 → 次
今 → 何時
今 → もっと
呼ぶ → 部屋
呼ぶ → 入る
見せる → 今日
近く → レストラン
禁煙 → 席
教える → 買う
教える → 場所
教える → 靴
教える → 荷物
教える → いう

A.1.3 Trigger pairs from CSJ

風 → 考える
無人島 → 持つ
持つ → 無人島
話 → 聞く
世紀 → 残す
住む → 町
本 → 読む
持つ → 考える
考える → 風
町 → 住む
考える → 持つ
印象 → 残る
そう → 考える
親 → 子供
とても → いい
本当に → 感じ
風 → 本当に
よい → 分かる
持つ → 風
子供 → 親
感じ → 本当に
本当に → 風
そう → 持つ
持つ → そう
本当に → 子供
風 → 持つ
考える → そう
分かる → 本当に
たばこ → 吸う
子供 → 本当に
持つ → 三つ
携帯 → 電話
どう → 考える
無人島 → 考える
本当に → 分かる
考える → まず
一番 → 大事
風 → やはり
海外 → 旅行
風 → 子供
子供 → 持つ
そう → やはり
聞く → 話
読む → 本
色々 → 考える
子供 → 風
どう → 風
無人島 → 三つ
本当に → とても
子供 → 子
持つ → まず
作る → 風
いい → 入る
やはり → そう
食べる → おいしい
入る → 本当に
おいしい → 食べる
入る → いい
持つ → 子供
アメリカ → 日本
本当に → 本当
作る → 食べる
感じ → 風
とても → 本当に
音楽 → 聞く
三つ → 無人島
犬 → 飼う
日本 → アメリカ
やはり → 風
本当に → 話
結構 → 感じ
考える → 無人島
結構 → 多い
家族 → 子供
考える → やはり
風 → どう
料理 → 作る
子供 → とても
作る → 持つ
考える → 一つ
本当に → 入る
使う → 作る
父 → 母
食べる → 作る
酒 → 飲む
絵 → 書く
いい → 子供
風 → まず
入る → 感じ
一番 → 最初
やはり → 考える
考える → 分かる
聞く → 本当に
考える → 子供
本当に → 持つ
風 → 一番
やはり → いい
仕事 → 会社
飼う → 犬
感じ → 結構

A.2 Trigger pairs extracted from initial transcriptions

This section provides two lists of trigger pairs extracted from the respective initial transcriptions of the two test sets used in chapter 4. Here, we compared the trigger pairs in each list with the trigger pairs extracted from the correct transcription of each test set. The trigger pairs in each list that were not present in the pairs extracted from each correct transcription were labeled as incorrect.

We can see that within the top 100 trigger pairs there are only two or three incorrect trigger pairs, respectively. This is because the erroneous words that form incorrect trigger pairs do not co-occur as frequently as correct words.

A.2.1 Trigger pairs from initial transcription of Sunday Discussion

ミス → マッチ
マッチ → ミス
構造 → 改革
不良 → 債権
債権 → 処理
公共 → 事業
産業 → 雇用
産業 → 建設
デフレ → スパイラル
産業 → 業界
借金 → 返済
公共 → 事業
不良 → 債権
やっぱり → 財源
ワークシェアリング → 日本
規模 → 借金
産業 → 創出
債権 → 不良
解雇 → ルール
債権 → 処理
国債 → 出す
銀行 → 廃止
できる → 話
ＧＤＰ → マイナス
マイナス → ＧＤＰ
デフレ → スパイラル
まず → 外交
事業 → 公共
考える → 産業
企業 → 不況
改革 → 景気
できる → 銀行
ことし → 政策 (incorrect)
不良 → 処理
不況 → 改革
政策 → 財政
借金 → 返済
骨太 → 方針
計画 → 見直す
基本 → 日本
改革 → 構造
アメリカ → 産業
日本 → 見る
出る → 不況
代表 → 日本 (incorrect)
公共 → 投資
強い → 国債
企業 → 銀行
成長 → 悪い
事業 → 公共
市場 → 供給
国債 → 多様
考える → 国債
改革 → 世界
話 → 銀行
分野 → 内閣
不況 → 原因
日本 → 中小
設備 → 投資
正社員 → 意味
整備 → 計画
政策 → 進める
借金 → 規模
構造 → 改革
一つ → できる
どんどん → 不況
お金 → 借りる
創出 → 産業
正社員 → 前向き
処理 → 債権
処理 → 銀行
産業 → 人間
景気 → スパイラル
一つ → 雇用
ルール → 解雇
スパイラル → 厳しい
そう → 雇用
不況 → 企業
出る → 処理
使う → どう
産業 → 近く
国家 → イスラム
構造 → 改革
雇用 → 対策
雇用 → 産業
ミス → マッチング
マイナス → スパイラル
できる → 出る
つくる → 必要
理由 → 船
不良 → 処理
働く → 構造
中期 → 来年度
所得 → 住宅
住宅金融公庫 → 住宅
失業 → 給付
示す → 国債
構造 → 成長
感覚 → 落ち込む
格下げ → 国債

A.2.2 Trigger pairs from initial transcription of National Diet

天然 → ガス
労働 → 監督
破壊 → 兵器
個人 → 閣僚
買う → 閣僚
官房 → 長官
基準 → 監督
イラク → 兵器
平壌 → 宣言
天然 → ガス
労働 → 基準
大量 → 破壊
北朝鮮 → そう
個人 → 消費
ガス → パイプライン
イラク → 協力
兵器 → 破壊
パイプライン → 方式
ガス → 天然
金融 → 大臣
サービス → 残業
宣言 → 北朝鮮
正常 → 交渉
記者 → 会見
エネルギー → 天然
イラク → 破壊
絶対 → もう
規制 → 改革
協力 → イラク
基準 → 労働
日本 → 来る
竹中 → 大臣
エネルギー → ガス
交渉 → 北朝鮮
ガス → 天然
国際 → 社会
ない → できる
兵器 → イラク
貯蓄 → 流れ
改革 → 規制
交渉 → できる
労働 → 状況
破壊 → 大量
申し上げる → 閣僚
いる → もう
日本 → できる
長官 → 官房
状況 → 労働
パイプライン → 陸上
できる → 北朝鮮
大臣 → 金融
正常 → 北朝鮮
解決 → 北朝鮮
イラク → 大臣 (incorrect)
消費 → 高齢
もる → 閣僚
ガス → エネルギー
やる → 北朝鮮
交渉 → 話
問題 → もう
未来 → 閣僚
破壊 → 廃棄
閣僚 → 竹中
イラク → 大量
兵器 → 廃棄
人達 → 人 (incorrect)
国債 → 個人
交渉 → 解決
いい → 閣僚
労働 → そう
報告 → 委員
パイプライン → 天然
労働 → 時間
天然 → エネルギー
残業 → サービス
インフラ → パイプライン
陸上 → 方式
個人 → 高齢
もう → 閣僚
日本 → そう
投資 → 流れ
パイプライン → インフラ
ガス → インフラ
もう → 買う
そう → できる
兵器 → 大量
人 → 被爆
久世 → 委員 (incorrect)
会見 → 個人
化学 → 兵器
安保理 → 国
人 → 受ける
イラク → 外務
いう → 北朝鮮
状況 → 監督
陸上 → パイプライン
長官 → 懇談
交渉 → 正常
大量 → 廃棄
国際 → そう

Bibliography

[1] Yuya Akita and Tatsuya Kawahara, "Language Model Adaptation based on PLSA of Topics and Speakers for Automatic Transcription of Panel Discussions," IEICE Transactions on Information and Systems, volume E88-D, number 3, pages 439–445, 2005.

[2] Lalit Bahl, Frederick Jelinek, and Robert L. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 5, number 2, pages 179–190, 1983.

[3] Jerome R. Bellegarda, "Exploiting Latent Semantic Information in Statistical Language Modeling," Proceedings of the IEEE, volume 88, number 8, pages 1279–1296, 2000.

[4] Brigitte Bigi, Armelle Brun, Jean-Paul Haton, Kamel Smaïli, and Imed Zitoumi, "A Comparative Study of Topic Identification on Newspaper and E-mail," Proceedings of the 8th International Symposium on String Processing and Information Retrieval, pages 238–241, 2001.

[5] Brigitte Bigi, Salma Jamoussi, and Kamel Smaïli, "Dynamic Topic Identification: Introduction of Trigger pairs in the Cache Model," Proceedings of the International Workshop on Speech and Computer, 2002.

[6] Simon Broman and Mikko Kurimo, "Methods for Combining Language Models in Speech Recognition," Proceedings of the European Conference on Speech Communication and Technology, pages 1317–1320, 2005.

[7] Peter F. Brown, Vincent J. Della Pietra, Peter V. de Souza, Jeniffer C. Lai, and Robert L. Mercer, "Class-Based n-gram Models of Natural Language," Computational Linguistics, volume 18, number 4, pages 467–479, 1992.

[8] Ciprian Chelba and Frederick Jelinek, "Recognition Performance of a Structured Language Model," Proceedings of the European Conference on Speech Communication and Technology, volume 4, pages 1567–1570, 1999.

[9] Stanley F. Chen, Kristie Seymore, and Ronald Rosenfeld, "Topic Adaptation for Language Modeling Using Unnormalized Exponential Models," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pages 681–684, 1998.

[10] Jen-Tzung Chien and Hung-Ying Chen, "Mining of Association Patterns for Language Modeling," Proceedings of the International Conference on Spoken Language Processing, pages 1369–1372, 2004.

[11] Philip R. Clarkson and Anthony J. Robinson, "Language Model Adaptation using Mixtures and an Exponentially Decaying Cache," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, volume II, pages 799–802, 1997.

[12] Philip R. Clarkson and Ronald Rosenfeld, "Statistical Language Modeling Using the CMU-Cambridge Toolkit," Proceedings of the European Conference on Speech Communication and Technology, volume 5, pages 2707–2710, 1997.

[13] Philip R. Clarkson, "The Applicability of Adaptive Language Modelling for the Broadcast News Task," Proceedings of the International Conference on Spoken Language Processing, volume 5, pages 1699–1702, 1998.

[14] Philip R. Clarkson, "Adaptation of Statistical Language Models for Automatic Speech Recognition," Ph.D. Thesis, University of Cambridge, 1999.

[15] Noah Coccaro and Daniel Jurafsky, "Towards Better Integration of Semantic Predictors in Statistical Language Modeling," Proceedings of the International Conference on Spoken Language Processing, volume 6, pages 2403–2406, 1998.

[16] Scott Deerwester, Susan T. Dumais, George W. Fumas, Thomas K. Landauer, and Richard Harshman "Indexing by Latent Semantic Analysis," Journal of the American Society of Information Science, volume 41, number 6, pages 391–407, 1990.

[17] Stephen Della Pietra, Vincent J. Della Pietra, Robert L. Mercer, and Salim Roukos, "Adaptive Language Modeling Using Minimum Discriminant Estimation," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, volume I, pages 633–636, 1992.

[18] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," Journal of the Royal Statistical Society, volume 39, number 1, pages 1–38, 1977.

[19] Ted Dunning, "Accurate Methods for the Statistics of Surprise and Coincidence," Association for Computational Linguistics, volume 19, number 1, pages 61–74, 1993.

[20] Marco Ferreti, Giulio Maltese, and Stefano Scarci, "Language Model and Acoustic Model Information in Probabilistic Speech Recognition," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pages 707–710, 1989.

[21] Radu Florian and David Yarowsky, "Dynamic Nonlocal Language Modeling via Hierarchical Topic-Based Adaptation," Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, pages 167–174, 1999.

[22] Daniel Gildea and Thomas Hofmann, "Topic-Based Language Models Using EM," Proceedings of the European Conference on Speech Communication and Technology, pages 2167–2170, 1999.

[23] Thomas Hofmann and Jan Puzicha, "Unsupervised Learning from Dyadic Data," Technical Report TR–98–042, 1998.

[24] Thomas Hofmann, "Probabilistic Latent Semantic Indexing," Proceedings of the Annual ACM Conference on Research and Development in Information Retrieval, pages 50–57, 1999.

[25] Xuedong Huang, Fileno Alleva, Hsiao-wuen Hon, Mei-Yuh Hwang, Kai-Fu Lee, and Ronald Rosenfeld, "The SPHINX-II Speech Recognition System: An Overview," Computer Speech and Language, volume 2, pages 137–148, 1993.

[26] Ryosuke Isotani, Shoichi Matsunaga, and Shigeki Sagayama, "Speech Recognition Using Function-Word N-Grams and Content-Word N-Grams," IEICE Transactions on Information and Systems, volume E78-D, number 6, pages 692–697, 1995.

[27] Rukmini Iyer and Mari Ostendorf, "Modeling Long Distance Dependence in Language: Topic Mixtures vs. Dynamic Cache Models," IEEE Transactions on Speech and Audio Processing, volume 7, number 1, pages 30–39, 1999.

[28] Edwin T. Jaynes, "Information Theory and Statistical Mechanics," Physical Review, volume 106, pages 620–630, 1957.

[29] Frederick Jelinek and Robert L. Mercer, "Interpolated Estimation of Markov Source Parameters from Sparse Data," Proceedings of the Workshop on Pattern Recognition in Practice, pages 381–397, 1980.

[30] Frederick Jelinek, "Self-Organized Language Modeling for Speech Recognition," Readings in Speech Recognition, pages 450–506, 1990.

[31] Frederick Jelinek, Bernard Merialdo, Salim Roukos, and M. Strauss, "A Dynamic Language Model for Speech Recognition," Proceedings of the DARPA Workshop on Speech and Natural Language, pages 293–295, 1991.

[32] Frederick Jelinek, "Statistical Methods for Speech Recognition," The MIT Press, 1997.

[33] Information-Technology Promotion Agency, Kyoto University, Nara Institute of Science and Technology, "Multipurpose Large Vocabulary Continuous Speech Recognition Engine Julius rev. 3.2," http://julius.sourceforge.jp/3.3/Julius-3.2-book-e.pdf

[34] Nobuhiro Kaji, Masashi Okamoto, and Sadao Kurohashi, "Paraphrasing Predicates from Written Language to Spoken Language Using the Web," Proceedings of the Human Language Technology Conference HLT-NAACL, pages 241–248, 2004.

[35] Slava M. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," IEEE Transactions on Acoustics, Speech and Signal Processing, volume 35, number 3, pages 400–401, 1987.

[36] Tatsuya Kawahara, C.-H. Lee, and Biing-Hwang Juang, "Flexible Speech Understanding Based on Combined Key-Phrase Detection and Verification," IEEE Transactions on Speech and Audio Processing, volume 6, number 6, pages 558–568, 1998.

[37] Tatsuya Kawahara, Akinobu Lee, Tetsunori Kobayashi, Kazuya Takeda, Nobuaki Minematsu, Shigeki Sagayama, Katsunobu Itou, Akinori Ito, Mikio Yamamoto, Atsushi Yamada, Takehito Utsuro, and Kiyohiro Shikano, "Free Software Toolkit for Japanese Large Vocabulary Continuous Speech Recognition," Proceedings of the International Conference on Spoken Language Processing, volume 4, pages 476–479, 2000.

[38] Tatsuya Kawahara, Hiroaki Nanjo, Takahiro Shinozaki, and Sadaoki Furui, "Benchmark Test for Speech Recognition Using the Corpus of Spontaneous Japanese," Proceedings of the ISCA and IEEE Workshop on Spontaneous Speech Processing and Recognition, pages 135–138, 2003.

[39] Tatsuya Kawahara, Akinobu Lee, Kazuya Takeda, Katsunobu Itou, and Kiyohiro Shikano, "Recent Progress of Open-Source LVCSR Engine Julius and Japanese Model Repository," Proceedings of the International Conference on Spoken Language Processing, pages 3069–3072, 2004.

[40] Sanjeev Khudanpur and Jun Wu, "A Maximum Entropy Language Model Integrating N-Grams and Topic Dependencies for Conversational Speech Recognition," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, volume I, pages 553–556, 1999.

[41] Sanjeev Khudanpur and Jun Wu, "Maximum Entropy Techniques for Exploiting Syntactic, Semantic and Collocational Dependencies in Language Modeling," Computer Speech and Language, volume 14, number 4, pages 355–372, 2000.

[42] Reinhard Kneser and Hermann Ney, "Improved Smoothing for n-gram Language Modeling," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pages 181–184, 1995.

[43] Roland Kuhn, "Speech Recognition and the Frequency of Recently Used Words: A Modified Markov Model for Natural Language," Proceedings of the International Conference on Computational Linguistics, pages 348–350, 1988.

[44] Roland Kuhn and Renato De Mori, "A Cache-Based Natural Language Model for Speech Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 12, number 6, pages 570–583, 1990.

[45] Roland Kuhn and Renato De Mori, "Corrections to A Cache-Based Natural Language Model for Speech Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 14, number 6, pages 691–692, 1992.

[46] Sadao Kurohashi and Manabu Ori, "Nonlocal Language Modeling Based on Context Co-occurrence Vectors," Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pages 80–86, 2000.

[47] Raymond Lau, Ronald Rosenfeld, and Salim Roukos, "Trigger-Based Language Models: A Maximum Entropy Approach," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, volume II, pages 45–48, 1993.

[48] Akinobu Lee, Tatsuya Kawahara, and Kiyohiro Shikano, "Julius – an Open Source Real-Time Large Vocabulary Recognition Engine," Proceedings European Conference on Speech Communication and Technology, volume 3, pages 1691–1694, 2001.

[49] Kikuo Maekawa, Hanae Koiso, Sadaoki Furui, and Hitoshi Isahara, "Spontaneous Speech Corpus of Japanese," Proceedings of the International Conference on Language Resources and Evaluation, volume 2, pages 947–952, 2000.

[50] Hiroaki Nanjo and Tatsuya Kawahara, "Language Model and Speaking Rate Adaptation for Spontaneous Presentation Speech Recognition," IEEE Transactions on Speech and Audio Processing, volume 12, number 4, pages 391–400, 2004.

[51] Thomas R. Niesler and Philip C. Woodland, "Modelling Word-Pair Relations in a Category-Based Language Model," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, volume 2, pages 795–798, 1997.

[52] Paul Placeway, Richard Schwartz, Pascale Fung, and Long Nguyen, "The Estimation of Powerful Language Models from Small and Large Corpora," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, volume II, pages 33–36, 1993.

[53] Lawrence Rabiner and Biing-Hwang Juang, "Fundamentals of Speech Recognition," Prentice Hall, 1993.

[54] Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil, "Inducing a Semantically Annotated Lexicon via EM-Based Clustering," Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, pages 104–111, 1999.

[55] Ronald Rosenfeld, "Adaptive Statistical Language Modeling: A Maximum Entropy Approach," Ph.D. Thesis CMU–CS–94–138, Carnegie Mellon University, 1994.

[56] Ronald Rosenfeld, "A Maximum Entropy Approach to Adaptive Statistical Language Modeling," Computer Speech and Language, volume 10, pages 187–228, 1996.

[57] Ronald Rosenfeld, "Two Decades of Statistical Language Modeling: Where do We Go from Here?," Proceedings of the IEEE, volume 88, number 8, pages 1270–1278, 2000.

[58] Ronald Rosenfeld, Stanley F. Chen, and Xiaojin Zhu "Whole-Sentence Exponential Language Models: a Vehicle for Linguistic-Statistical Integration," Computer Speech and Language, volume 15, number 1, pages 55–73, 2001.

[59] Gerard Salton, "Developments in Automatic Text Retrieval," Science, volume 253, pages 974–980, 1991.

[60] Ruhi Sarikaya, Agustin Gravano, and Yuqing Gao, "Rapid Language Model Development Using External Resources for New Spoken Dialog Domains," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, volume 1, pages 573–576, 2005.

[61] Richard Schwartz and Yen-Lu Chow, "The N-best Algorithm: Efficient and Exact Procedure for Finding the N Most Likely Sentence Hypotheses," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pages 81–84, 1990.

[62] Satoshi Sekine, John Sterling, and Ralph Grishman, "NYU/BBN 1994 CSR Evaluation," Proceedings of the Spoken Language Systems Technology Workshop, pages 148–152, 1995.

[63] Satoshi Sekine, "Modeling Topic Coherence for Speech Recognition," Proceedings of the International Conference on Computational Linguistics, pages 913–918, 1996.

[64] Satoshi Sekine and Ralph Grishman, "NYU Language Modeling Experiments for the 1995 CSR Evaluation," Proceedings of the Spoken Language Systems Technology Workshop, pages 123–128, 1996.

[65] Satoshi Sekine, Andrew Borthwick, and Ralph Grishman, "NYU Language Modeling Experiments for the 1996 CSR Evaluation," Proceedings of the DARPA Speech Recognition Workshop, 1997.

[66] Kristie Seymore and Ronald Rosenfeld, "Using Story Topics for Language Model Adaptation," Proceedings of the European Conference on Speech Communication and Technology, volume 4, pages 1987–1990, 1997.

[67] Kristie Seymore and Ronald Rosenfeld, "Large-Scale Topic Detection and Language Model Adaptation," Technical Report CMU–CS–97–152, 1997.

[68] Kristie Seymore, Stanley F. Chen, and Ronald Rosenfeld, "Nonlinear Interpolation of Topic Models for Language Model Adaptation," Proceedings of the International Conference on Spoken Language Processing, volume 6, pages 2503–2506, 1998.

[69] Tohru Shimizu, Hirofumi Yamamoto, Hirokazu Masataki, Shoichi Matsunaga, and Yoshinori Sagisaka, "Spontaneous Dialogue Speech Recognition Using Cross-Word Context Constrained Word Graph," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, volume 1, pages 145–148, 1996.

[70] Andreas Stolcke and Elizabeth Shriberg, "Statistical Language Modeling for Speech Disfluencies," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, volume 1, pages 405–408, 1996.

[71] Toshiyuki Takezawa, Eiichiro Sumita, Fumiaki Sugaya, Hirofumi Yamamoto, and Seiichi Yamamoto, "Toward a Broad-Coverage Bilingual Corpus for Speech Translation of Travel Conversations in the Real World," Proceedings of the International Conference on Language Resources and Evaluation, volume 1, pages 147–152, 2002.

[72] Yik-Cheung Tam and Tanja Schultz, "Dynamic Language Model Adaptation using Variational Bayes Inference," Proceedings of the European Conference on Speech Communication and Technology, volume 1, pages 5–8, 2005.

[73] Christoph Tillmann and Hermann Ney, "Selection Criteria for Word Trigger Pairs in Language Modeling," Proceedings of the International Colloquium on Grammatical Inference, pages 95–106, 1996.

[74] Christoph Tillmann and Hermann Ney, "Word Triggers and the EM Algorithm," Proceedings of the ACL Special Interest Group Workshop on Computational Natural Language Learning, pages 117–124, 1997.

[75] Jun Wu and Sanjeev Khudanpur, "Combining Nonlocal, Syntactic and N-Gram Dependencies in Language Modeling," Proceedings of the European Conference on Speech Communication and Technology, volume 5, pages 2179–2182, 1999.

[76] Jun Wu and Sanjeev Khudanpur, "Syntactic Heads in Statistical Language Modeling," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pages 1699–1702, 2000.

[77] Jun Wu and Sanjeev Khudanpur, "Building a Topic-Dependent Maximum Entropy Model for Very Large Corpora," Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, volume 1, pages 777–780, 2002.

[78] Jun Wu, "Maximum Entropy Language Modeling with Non-Local Dependencies," Ph.D. Thesis, Johns Hopkins University, 2002.

[79] Peng Xu and Lidia Mangu, "Using Random Forest Language Models in the IBM RT-04 CTS System," Proceedings of the European Conference on Speech Communication and Technology, pages 741–744, 2005.

[80] Guo-Dong Zhou and Kim-Teng Lua, "Word Association and MI-Trigger-based Language Modeling," Proceedings of the 36th conference on Association for Computational Linguistics, pages 1465-1471, 1998.

Publications

[1] C. Troncoso and T. Kawahara: "Trigger-Based Language Model Adaptation for Automatic Transcription of Panel Discussions," IEICE Trans. on Information and Systems, vol.E89-D, no.3, pp.1024–1031 (Mar. 2006).

[2] Y. Akita, C. Troncoso, and T. Kawahara: "Automatic Transcription of Meetings Using Topic-Oriented Language Model Adaptation," to appear in Proc. of WESPAC 2006 (Jun. 2006).

[3] C. Troncoso and T. Kawahara: "Trigger-Based Language Model Adaptation for Automatic Meeting Transcription," Proc. of Interspeech-Eurospeech 2005, pp.1297–1300 (Sep. 2005).

[4] C. Troncoso, T. Kawahara, H. Yamamoto, and G. Kikui: "Trigger-Based Language Model Construction by Combining Different Corpora," Proc. of the Pacific Association for Computational Linguistics (PACLING) 2005, pp.340–344 (Aug. 2005).

[5] C. Troncoso and T. Kawahara: "Automatic Transcription of Panel Discussions Using Trigger-Based Language Model Adaptation," IPSJ SIG Technical Report, 2005–SLP–57–3 (Jul. 2005).

[6] C. Troncoso, T. Kawahara, H. Yamamoto, and G. Kikui: "Trigger-Based Language Model Construction by Combining Different Corpora," IEICE Technical Report, SP2004–100 (Dec. 2004).

[7] C. Troncoso and T. Kawahara: "Enhancement to Initial Transcription-Based Trigger Language Model Adaptation," Proc. of the Autumn Meeting of the Acoustical Society of Japan 2005, 2–1–2 (Sep. 2005).

[8] C. Troncoso and T. Kawahara: "Trigger-Based Language Model Adaptation for Automatic Transcription of Panel Discussions," Proc. of the Spring Meeting of the Acoustical Society of Japan 2005, 1–5–22 (Mar. 2005).

[9] C. Troncoso, H. Yamamoto, and G. Kikui: "Trigger-Based Language Model Adaptation Using Two Different Corpora," Proc. of the Autumn Meeting of the Acoustical Society of Japan 2004, 2–1–11 (Sep. 2004).

[10] C. Troncoso, S. Matsuda, M. Nakai, H. Shimodaira, and K. Torisawa: "An Extension to the Trigger Language Model Based on a Probabilistic Thesaurus," Proc. of the Spring Meeting of the Acoustical Society of Japan 2003, 3–4–7 (Mar. 2003).

[11] C. Troncoso, H. Yamamoto, and G. Kikui: "Trigger-Based Language Model Adaptation Using Two Different Corpora," ATR Technical Report, TR–SLT–0074 (May 2004).

[12] C. Troncoso: "An Extension to the Trigger Language Model Based on a Probabilistic Thesaurus and Document Clusters," Master's Thesis, Japan Advanced Institute of Science and Technology (March 2003).