論文 / 著書情報
Article / Book Information

Title	Language Model Acquisition from a Text Corpus for Speech Understanding
Author	Tatsuo Matsuoka, Robert Hasson, Michael Barlow, Sadaoki Furui
Journal/Book name	IEEE ICASSP 1996, Vol. 1, No. , pp. 413-416
発行日 / Issue date	1996, 5
権利情報 / Copyright	(c)1996 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

LANGUAGE MODEL ACQUISITION FROM A TEXT CORPUS FOR SPEECH UNDERSTANDING

Tatsuo Matsuoka, Robert Hasson[†], Michael Barlow, and Sadaoki Furui

NTT Human Interface Laboratories
3-9-11 Midori-cho, Musashino-shi, Tokyo 180, JAPAN
[†] on leave from EURECOM Institute, FRANCE
matsuoka@splab.hil.ntt.jp

ABSTRACT

Speech understanding can be viewed as a problem of translating input natural language of speech recognition results into output semantic language. This paper describes automatic acquisition of a language model for translating natural language into semantic language from a text corpus using a stochastic method. The method estimates co-occurrence probabilities of input and output grammar rules as a translation language model. Since the amount of texts is limited, estimating a reliable language model is difficult. Therefore, we propose a method of concisely modeling input and output grammars in order to estimate a reliable translation model. Our method is shown to be effective by experiments using the ARPA ATIS task.

1. INTRODUCTION

A speech understanding system requires two functions: one is speech recognition, which converts speech input into a sequence of words, and the other is language processing, which extracts meaning from the word sequence. As for speech recognition, we previously reported on the N-best search algorithm, which uses triphone context-dependent acoustic models for intra- and inter-word contexts[(1)]. This recognizer achieved a word-error rate of 5.2% for the ATIS task.

This paper describes language processing which converts the natural language of speech recognition results into semantic language. Usually, language processing has been implemented by writing grammar rules manually. However, this takes a considerable amount of time and effort, and the grammar cannot easily transfered to other tasks. Therefore, automatic acquisition of a language model or grammar rules is strongly required. This paper describes a stochastic method of automatically acquiring a language model from a corpus.

2. SPEECH UNDERSTANDING SYSTEM

Figure 1 illustrates our configuration of a speech understanding system. The system consists of a speech recognition module and a language processing module. This paper focuses on the language processing module.

The language processing module translates the speech recognition results into a database inquiry language. Although SQL is the database inquiry language for ATIS, a WIN (Wizard Input) sentence, which can be directly translated into an SQL inquiry, is available for each sentence in the ATIS corpus, so we used WIN sentences as the semantic language for our experiments.

3. STOCHASTIC TRANSLATION LANGUAGE MODELING

In the area of machine translation, a stochastic approach for machine translation was suggested about 50 years ago. At that time, however, there were few machine-readable text database, and computer performance was too poor to implement the stochastic approach.

In 1990, Brown et al. proposed a basic framework of stochastic language modeling for machine translation[(2)]. Brown et al. defined a translation language model as

$$P(\mathrm{e} \mid \mathrm{f}) = \frac{P(\mathrm{e})P(\mathrm{f} \mid \mathrm{e})}{P(\mathrm{f})}\ , \qquad (1)$$

where e is an English sentence and f is a French sentence. One can translate French into English by finding

$$\hat{\mathrm{e}} = \arg\max_{\mathrm{e}} P(\mathrm{e})P(\mathrm{f} \mid \mathrm{e})\ . \qquad (2)$$

The language model $P(\mathrm{e})$ is estimated using English texts. The translation language model $P(\mathrm{f} \mid \mathrm{e})$ is estimated using a parallel text corpus, which consists of French and English sentences. They reported that $P(\mathrm{e})P(\mathrm{f} \mid \mathrm{e})$ can achieve better translation than $P(\mathrm{e} \mid \mathrm{f})$, because $P(\mathrm{e})$ results in sentences that consists of more natural English[(4)]. In their experiment using the Hansard corpus, they set a vocabulary size of 58K for French and

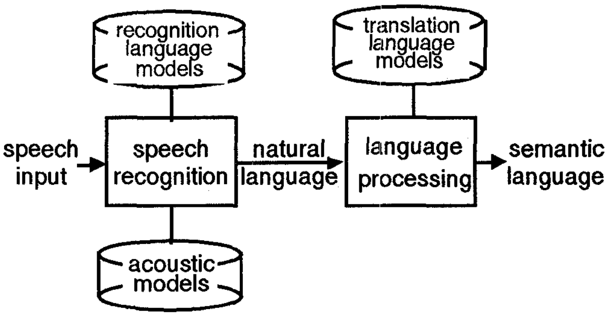

Figure 1 Speech understanding system

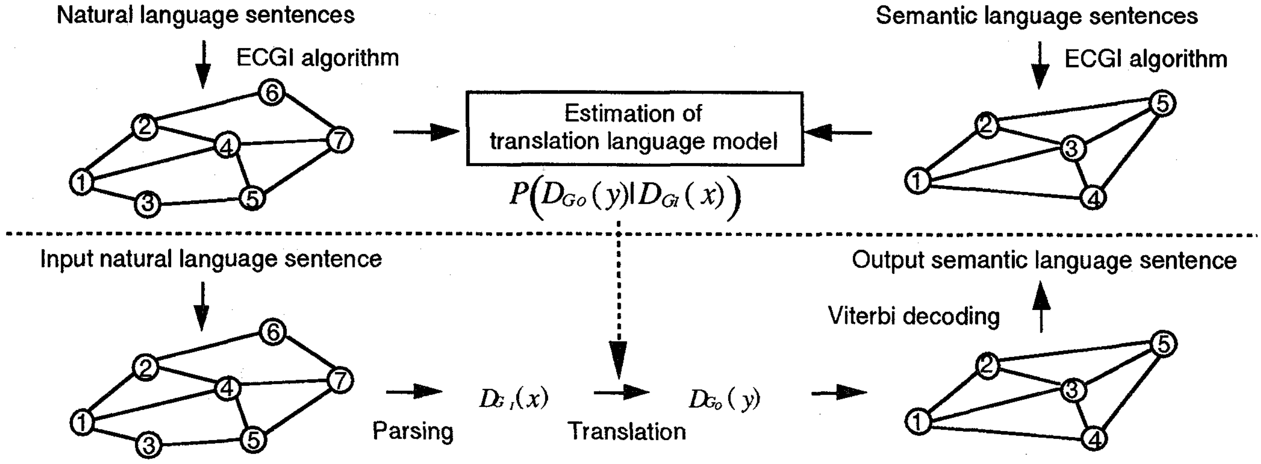
Natural language sentences
ECGI algorithm
Estimation of translation language model
$P(D_{Go}(y)|D_{Gi}(x))$
Semantic language sentences
ECGI algorithm
Input natural language sentence
Output semantic language sentence
Viterbi decoding
$D_{Gi}(x)$
$D_{Go}(y)$
Parsing
Translation

Figure 2 Estimation of the translation language model and translation using the model

41K for English, and used 1.7M pairs of sentences to estimate the translation language model. The method was tested using 100 sentences which were not included in the training set. They reported that 60 of the 100 sentences were acceptably translated[3]. Although they used many pairs of sentences, the estimation may have been inefficient because they estimated the translation language model $P(\mathrm{e})P(\mathrm{f}\,|\,\mathrm{e})$ directly from raw texts which are sometimes redundant or ambiguous.

Pieraccini et al. proposed a method that extracts the semantic structure from a sentence by using a stochastic model[5,6]. This method converts the word sequence of speech recognition results into a sequence of concepts. They formulated a speech understanding problem to find $\hat{W}$ and $\hat{C}$ in

$$P(\hat{W},\hat{C}\,|\,A)=\max_{W\times C}P(W,C\,|\,A)\ , \qquad (3)$$

where A is acoustic observation of speech, W is a word sequence, and C is a sequence of concepts. Since

$$P(W,C\,|\,A)=\frac{P(A\,|\,W,C)P(W\,|\,C)P(C)}{P(A)} \qquad (4)$$

holds, they estimated $P(A\,|\,W,C)\approx P(A\,|\,W)$ as an acoustic model, and $P(W\,|\,C)P(C)$ as a concept language model. The concept language model is an HMM, in which word sequences are observation vectors and sequences of concepts correspond to a hidden state sequence.

Vidal et al. extended this idea to translation from natural language into semantic language replacing concepts with grammar rules[7]. They first generated grammar rules for input natural language and output semantic language from the training texts using the Error Correcting Grammar Inference (ECGI) algorithm[8], then estimated the conditional probabilities of the grammar rules of the input and output language. For practical applications, however, input natural language varies widely, and the number of grammar rules becomes very large. This makes it difficult to accurately estimate a translation language model because the matrix of the conditional probabilities becomes sparse.

This paper proposes a method of coping with the sparseness of the conditional probabilities. We incorporate the context-free grammar inference algorithm proposed by McCandless et al.[9] into the estimation of the translation language model. Using this grammar inference, we can reduce the number of states in each grammar network, because similar words are merged into non-terminal symbols.

4. LANGUAGE PROCESSING FOR SPEECH UNDERSTANDING

Figure 2 illustrates an estimation of a stochastic translation language model, and also shows a translation using the model. The upper figure shows the estimation of the language model. Using input natural language and output semantic language, grammar networks are generated by using the ECGI algorithm. Then, for each pair of natural and semantic language sentences, sequences of grammar rules are obtained and conditional probabilities are calculated. The lower figure shows the process of translating a natural language sentence into a semantic language sentence. The input sentence is parsed and a sequence of grammar rules is derived. Then the translation language model is applied and a sequence of output grammar rules is found. The output semantic language sentence is determined by the Viterbi search in the output grammar network; this sentence is the best path for the sequence of grammar rules.

4.1 Generation of a grammar network

Grammar networks for input and output languages are generated using the ECGI algorithm. The ECGI algorithm parses sentences one by one from the training set, and adds necessary states and transitions to the existing grammar network to parse the input sentence. The best alignment between the input sentence and the grammar network is determined using the Error Correcting Parsing (ECP) algorithm[9]. New states and/or transitions are added along with the best path. The Levenstein distance defined by the Eq. (5) is used as the distance measure for ECP alignment.

$$d(X,Y) = \min_{s}(p \cdot sub_s + q \cdot ins_s + r \cdot del_s) \qquad (5)$$

Here, p, q, and r are weighting factors, s is the path in the grammar network, sub_s is the substitution error, ins_s is the insertion error, and del_s is the deletion error.

The number of states in the ECGI-derived grammar network depends on the order in which sentences are presented in the ECGI algorithm. If the longer sentences (in terms of number of words) are given first, the shorter sentences can take advantage of existing grammar states generated by the longer sentences, thus the number of states becomes relatively small.

4.2 Estimation of translation language model

We estimate the conditional probabilities of the input and output grammar rules by using input-output pairs of training sentences.

Let G_I and G_O be the input and output grammars. The problem is to find $\hat{y}$ that satisfies Eq. (6).

$$\hat{y} = \underset{y \in L(G_o)}{\arg\max}\ P(y \mid x) \qquad (6a)$$

$$= \underset{y \in L(G_o)}{\arg\max}\ P(x \mid y)P(y) \qquad (6b)$$

Sentences x and y can be represented as sequences of grammar rules $D_{G_I}(x)$ and $D_{G_O}(y)$.

$$D_{G_I}(x) = \left\{ r_{I1}^x, r_{I2}^x, \ldots, r_{In}^x \mid r_{Ii}^x \in G_I \right\} \qquad (7)$$

$$D_{G_O}(y) = \left\{ r_{O1}^y, r_{O2}^y, \ldots, r_{Om}^y \mid r_{Oi}^y \in G_O \right\} \qquad (8)$$

If G_I and G_O are unambiguous grammars, $D_{G_I}(x)$ and $D_{G_O}(y)$ are uniquely found. Otherwise, $D_{G_I}(x)$ and $D_{G_O}(y)$ can be approximated using the Viterbi algorithm. Eq. (6) can thus be rewritten as

$$\hat{y} = \underset{y \in L(G_o)}{\arg\max}\ P\left(D_{G_O}(y) \mid D_{G_I}(x) \right) \qquad (9a)$$

$$= \underset{y \in L(G_o)}{\arg\max}\ P\left(D_{G_I}(x) \mid D_{G_O}(y) \right) P(y) \ . \qquad (9b)$$

Vidal et al. defined $P\left(D_{G_I}(x) \mid D_{G_O}(y) \right)$ as

$$P\left(D_{G_I}(x) \mid D_{G_O}(y) \right)$$

$$\approx \prod_{r_O \in D_{G_O(y)}} \left(\prod_{r_I \in D_{G_I(x)}} P(r_I \mid r_O) \prod_{r_I \notin D_{G_I(x)}} P(not\ \ r_I \mid r_O) \right), \quad (10)$$

and estimated the conditional probabilities based on Eq. (9b).

There, $P(not\ \ r_I \mid r_O)$ is the probability of *not* using r_I in the input derivation, given that r_O is used in the output derivation. When the grammars are very small, using $P(not\ \ r_I \mid r_O)$ is effective. However, our grammar is roughly three times larger than Vidal's grammar. For practical grammars, evaluating irrelevant co-occurrences along with relevant co-occurrences can contaminate the effectiveness of the conditional probabilities. It also makes the computation inconvenient because the product of the probabilities becomes very small as the number of terms increases. Threfore, we decided not to use $P(not\ \ r_I \mid r_O)$.

Since there are usually fewer grammar rules for semantic language than for natural language, we estimated $P\left(D_{G_O}(y) \mid D_{G_I}(x) \right)$ based on Eq. (9a), instead of estimating $P\left(D_{G_I}(x) \mid D_{G_O}(y) \right)$ based on Eq. (9b). Thus, $P\left(D_{G_O}(y) \mid D_{G_I}(x) \right)$ is estimated as

$$P\left(D_{G_O}(y) \mid D_{G_I}(x) \right) \approx P\left(r_O \mid r_{I1}^x, r_{I2}^x, \ldots, r_{In}^x \right)$$

$$\approx \frac{N(r_O, x)}{N(x)}, \qquad (11)$$

where $N(x)$ is the number of sentences that G_I can generate. It is impossible to calculate $N(x)$ in practice, so we approximate the probability as

$$\hat{P}\left(r_O \mid r_{I1}^x, r_{I2}^x, \ldots, r_{In}^x \right)$$

$$\approx \frac{P^{\alpha}(r_O) \prod_{r_I \in D_{G_I}(x)} P^{\beta}(r_O \mid r_I)}{\sum_{\substack{r\ \text{with the same} \\ \text{initial state as}\ r_O}} \left[P^{\alpha}(r) \prod_{r_I \in D_{G_I}(x)} P^{\beta}(r \mid r_I) \right]} \ . \qquad (12)$$

When there are many grammar rules, estimating $P\left(D_{G_O}(y) \mid D_{G_I}(x) \right)$ is difficult because the probability parameter space becomes sparse as the number of grammar rules increases. Therefore, the number of grammar rules must be decreased.

4.3 Grammar state reduction using context-free grammar inference

McCandless et al. proposed a context-free grammar inference algorithm[9]. This algorithm generates grammar rules in a bottom-up manner by using word bigram probabilities as the distance measure. The distance measure between words or non-terminal symbols is defined as follows.

$$\left\| u_i, u_j \right\| = d(P_i, P_j) + d(P_j, P_i) \qquad (13)$$

$$d(P_i, P_j) = \sum_{C \in Context} P_i(C) \times \log \frac{P_i(C)}{P_j(C)} \qquad (14)$$