SELECTIVE BACK-OFF SMOOTHING FOR INCORPORATING GRAMMATICAL CONSTRAINTS INTO THE N-GRAM LANGUAGE MODEL

Tomoyosi AKIBA†, Katunobu ITOU‡, Atsushi FUJI, Tetsuya ISHIKAWA†

† National Institute of Advanced Industrial Science and Technology (AIST)
1-1-1 Umezono, Tsukuba, 305-8568, JAPAN, E-mail: t-akiba@aist.go.jp
‡ University of Library and Information Science
1-2 Kasuga, Tsukuba, 305-8550, JAPAN

ABSTRACT

Spoken queries submitted to question answering systems usually consist of query contents (e.g. about newspaper articles) and frozen patterns (e.g. WH-words), which can be modeled with N-gram models and grammar-based models, respectively. We propose a method to integrate those different types of models into a single N-gram model. We represent the two types of language models in a single word network. However, common smoothing methods, which are effective for N-gram models, decrease grammatical constraints for frozen patterns. For this problem, we propose a selective back-off smoothing method, which controls a degree to which smoothing is applied depending on the network fragment. Additionally, resulting models are compatible with the conventional back-off N-gram models, and thus existing N-gram decoders can easily be used. We show the effectiveness of our method by way of experiments.

1. INTRODUCTION

The N-gram model has been used successfully as a language model for large vocabulary continuous speech recognition (LVCSR) systems. The N-gram model is simple but robust enough to model all word sequences in the vocabulary. However, it needs a large training corpus and such a corpus cannot be easily constructed unless there already exists a large text corpus based on, for example, newspaper articles.

On the other hand, the grammar-based model is used as a language model for tasks involving a relatively small vocabulary. This model does not need a training corpus because it takes advantage of linguistic knowledge. It can model correlations more distant than is possible with the N-gram model, which can model only local relations between words. Thus, some spoken sentences can be modeled more suitably by the N-gram models and some more suitably by the grammar-based model. This is also true from an intra-sentence perspective – some parts of sentence are best modeled by N-gram model and some parts are best modeled by a grammar-based model.

For example, question answering systems receive queries that often consist of a part that conveys various query contents about, for example, newspaper articles, and a part that represents a frozen pattern for query sentences. The first part is best dealt with by using an N-gram model trained with a corpus of newspaper articles, and the second part is best dealt with by using the grammar-based model.

![Fig. 1. Integrated Word Network](image)

Another example is phrases that cannot be predicted from past training data, such as those concerning telephone numbers, ID numbers, a date, a time for a reservation, etc. Recognizing such a phrase is often necessary to achieve the given task. Those phrases seem to be best dealt with by using the grammar-based model while the other parts of the sentence is dealt with by using an N-gram model.

In this paper, we will explain how these two types of models can be integrated into a single N-gram model. The key idea is that we assume the grammar is described in regular language, so we should be able to represent it in an N-gram which is also equal to regular language. The problem is that the grammatical constraint is incompatible with smoothing, because the grammar tries to assign a zero probability to non-grammatical connections of the words while smoothing tries to assign a non-zero probability to avoid the zero frequency problem. To solve this inconsistency, we have developed a selective back-off smoothing method.

2. INTEGRATION OF THE CONVENTIONAL N-GRAM AND GRAMMAR-BASED MODELS

The sentences modeled by a conventional N-gram model can be expressed as a fully connected word network, in which a word can be followed by any word in the vocabulary of the model. On the other hand, the sentences modeled by a grammar-based model can be expressed as a partially connected word network, in which a word can be followed by only specific words according to the grammar. Integration of

---

1This is a reasonable assumption because any language of finite length is known to be included in regular language. Moreover, an algorithm exists to approximate CFGs into finite state automata [1].

The second and third authors are also members of CREST, Japan Science and Technology Corporation.
the conventional N-gram model (we will refer to it as “base N-gram”) and the grammar-based model can be achieved by joining the two different types of word networks so that any word in the base N-gram might be followed by the beginning words of the grammar and the ending words of the grammar might be followed by any word of the base N-gram (Fig. 1).

In the integrated word network, the N-gram probabilities are assigned so that the N-gram probability that runs along the directed arcs of the network might have a positive value and the other probabilities might be zero. With such a probability distribution, the integrated model is able to convey the characteristics of both the N-gram and the grammar.

To obtain such a distribution, we must give the N-gram counts along the arcs of the network, then calculate the probabilities. The next section of this paper will show how to construct the word network for the grammar and how to obtain the N-gram counts on the network. In section 4, we will introduce the selective back-off smoothing method that is used calculate the probabilities while preserving the characteristics of both the N-gram and the grammar.

3. GRAMMAR-BASED LANGUAGE MODELING

3.1. Word Network

Though there are many way to describe regular language, we will use a word network that consists of nodes corresponding to words and directed arcs corresponding to word-to-word transitions.

Such a network can be easily obtained for given example sentences. For example, we can obtain the network at the left of Fig. 2 from the following three Japanese sentences that denote questions about the date.

\[ naN / neN / desu / ka \]
\[ naN / neN / naN / gatsu / desu / ka \]
\[ naN / gatsu / naN / nichi / desu / ka \]

using the words “naN” (what), “neN” (year), “gatsu” (month), “nichi” (day), “desu” (is) and “ka” (interrogative). From the three sentences, we can get neighboring word pairs as follows.

\[ A = \{(naN, neN), (naN, gatsu), (neN, naN), (gatsu, naN), (naN, desu), (gatsu, desu)\} \]

Fig. 2. Word Network

If we assume word-to-word transition is possible if the word pair appear in A, the word network \( G_1 \) is defined by 4-tuples \((W_A, W_B, W_E, A)\) where \( W_A, W_B \) and \( W_E \) are, respectively, the set of all words, the set of beginning words, the set of ending words of the network. In this case,

\[ W_A = \{naN neN gatsu nichi desu ka\} \]
\[ W_B = \{naN\} \]
\[ W_E = \{ka\} \]

Unfortunately, the simple network \( G_1 \) fails to model the Japanese query sentences about the date correctly, because the sentences below that can be modeled by \( G_1 \) are never used in practice.

* naN / neN / naN / neN / desu / ka
  (the word “neN (year)” is repeated)
* naN / gatsu / naN / neN / desu / ka
  (a more specific word “gatsu (month)” comes before a less specific word “neN (year)”)

To exclude such ill-formed sentences, the word network should be improved by separating the word nodes according to their context (as shown at the right of Fig. 2). With the new symbol assigned to the separated nodes, the improved word network \( G_2 \) is described by \((W_A', W_B, W_E, A')\) where:

\[ W_A' = W_A \cup \{naN n-1\} \]
\[ A' = \{(naN, neN), (naN, gatsu), (naN, nichi)\} \]

In this way, introducing the new nodes allow word networks to express any long distance dependence between words.

To integrate the word network with the base N-gram, two requirements must be met when constructing the word network. First, the vocabulary for the network must be distinguished from that for the base N-gram. This can be easily achieved by applying different word symbols to the same words in the network and the base N-gram. Namely, we gave “@” as the prefix for word symbols in the network.

Second, the beginning words must not arrive other than at the beginning of the network and the ending words must not arrive other than at the ending of the network. In other words, the word set \( W_A \) consists of exclusive word sets \( W_B, W_I \) and \( W_E \) (i.e. \( W_A = W_B \cup W_I \cup W_E \)), that respectively correspond to beginning words, intermediate words, and ending words, i.e. \( W_B \cap W_I = \emptyset \cap W_I \cap W_E = \emptyset \cap W_B \cap W_E = \emptyset \).

3.2. Calculating Probability with Extra N-gram Counts

To give the N-gram probabilities of the words in the word network, we need the N-gram counts both on the word network and on the bridge connecting the network with the base N-gram.

The conventional N-gram counts are consistent and complete. To calculate the N-gram probability, only the longest N-gram counts \( C(w_{i-N+1}^i) \) (where \( w_{i-N+1}^i \) denotes the word sequence \( w_{i-N+1} \cdots w_{i-1} w_i \)) are needed because the shorter N-gram counts are obtained from them by recursively summing them up giving all the last (or first) words. Consequently, an N-gram count \( C(w_{i-N+1}^i) \) is used to calculate all the probabilities that predict the words \( w_{i-N+1} \cdots w_i \).
In other words, raising an N-gram count \( C(w_i|w_{i-N+1}) \) results in raising all the probabilities used to predict the words \( w_{i-N+1} \cdots w_i \).

If we provide an arbitrary number of extra N-gram counts of an arbitrary length, the consistency and the completeness of the conventional N-gram counts are broken. Furthermore, we want to give the extra N-gram counts \( C(w_i|w_{i-N+1}) \) only to raise the probability of the last word \( w_i \). Thus, we need to redefine the usage of the N-gram counts as follows.

- The N-gram counts \( C_1(w_i), C_2(w_{i-1}), \cdots, C_N(w_{i-N+1}) \) are given separately according to their length 1 \( \cdots \) N.
- Each N-gram count \( C_n(w_{i-n+1}) \) is used only to calculate the probability used to predict the last word \( w_i \).

For example, the probability estimated through the maximum likelihood method can be calculated from the conventional N-gram counts as

\[
P_{ML}(w_i|w_{i-N+1}) = C_n(w_{i-n+1})/\sum_{w_j} C_n(w_{i-n+1})
\]

which relies on consistency among the counts. On the other hand, with the extra N-gram counts, the probability must be calculated as

\[
P_{ML}(w_i|w_{i-N+1}) = C_n(w_{i-n+1})/\sum_{w_j} C_n(w_{i-n+1})
\]

### 3.3. Providing N-gram Counts onto the Word Network

To provide N-gram counts to the constructed word network, we used the existing counts on the base N-gram, though several approaches are applicable that include giving equal counts at each branch of the network. To do so, the words in the network must be of the same unit with the base N-gram.\(^2\)

The N-gram counts related to the network can then be obtained from the corresponding counts in the base N-gram. We obtain the N-gram count \( C_n(w_{i-n+1}) \) in the network by copying the count \( C_n(w_{i-n+1}) \) found in the base N-gram. We also need the counts corresponding to the bridge between the network and the base N-gram. Such counts can also be obtained by copying the corresponding counts in the base N-gram. Namely, both \( C_n(w_{i-n+1}) \) and \( C_n(w_{i-n+1}) \) are obtained from \( C_n(w_{i-n+1}) \). Among these, we obtain \( C_n(w_{i-n+1}) \) by multiplying \( C_n(w_{i-n+1}) \) by the weight \( \gamma \). \( \gamma \) is introduced so that we can assign a higher probability to a sentence that traces the network and \( \gamma \) is assumed to be given a value not less than 1 (Fig. 3).

\[
P(w_i|w_{i-1}) = \left\{ \begin{array}{ll}
d'_{w_{i-1}}P_{ML}(w_i|w_{i-1}) & C_2(w_{i-1}) > 0 \\
0 & C_2(w_{i-1}) = 0 \end{array} \right.
\]

As a result, we get

\[
P(w_i) = 0 \quad \text{if} \quad w_i \in W_I \cup W_E
\]

\[
\alpha_1(w_{i-1}) = 0 \quad \text{if} \quad w_{i-1} \in W_B \cup W_I
\]

The model to which the selective back-off smoothing is applied has two significant features.

- The probability \( P(w_i|w_{i-N+1}) \), where \( w_i \in W_I \cup W_E \) and the word sequence \( w_{i-N+1} \) is not allowed by the grammar, becomes zero.

In this case, the N-gram counts \( C_n(w_{i-n+1}) \) for \( n > 1 \) are zero. Thus the probability is “back-off”ed to the uni-gram:

\[
P(w_i|w_{i-N+1}) = \alpha_{N-1} \cdots \alpha_2 \cdot \alpha_1(w_{i-1}) P(w_i) = 0
\]

\(^2\)The definition of the word unit is important and should be done beforehand to model Japanese because words are not explicitly separated by spaces in Japanese sentences.

\(^3\)More precisely, the uni-gram discounting must be applied to only the word set \( W_I \cup W_E \) because the uni-gram probabilities on \( W_I \cup W_E \) are set to 0 by Equation 2.
Fig. 4. Selective Back-off Smoothing

Fig. 5. Word network used for the experiment

because $P(w_i) = 0$ from Equation 3.

Especially, the probabilities used to predict the word $w \in W_F \cup W_E$ from the words (in the base N-gram) $w \in W_U$ are always zero.

- The probabilities used to predict the word (in the base N-gram) $w \in W_I$ from the words (in the network grammar, but excepting the endings) $w \in W_B \cup W_I$ are always zero.

Also in this case, the N-gram counts $C_n(w_{i-n+1}^i)$ for $n > 1$ are zero. Thus, the probability is “back-off”ed to the uni-gram:

$$P(w_i | w_{i-N+1}^{i-1}) = \alpha_{N-1} \cdot \cdots \cdot \alpha_2 \cdot \alpha_1(w_{i-1})P(w_i) = 0$$

because the context word $w_{i-1}$ is in $W_B \cup W_I$ and we get $\alpha(w_{i-1}) = 0$ from Equation 4.

The resulting integrated model is compatible with conventional back-off N-gram models so that it can work as the language model for existing LVCSR systems.

5. EXPERIMENTAL RESULTS

We developed a word network for the Japanese frozen patterns used for Question Answering [3] (Fig.5). We also prepared a base N-gram of 20,000 words that were obtained from newspaper articles collected over 111 months. We then integrated these two models according to our method explained above (referred to as the base+net model). To enable comparison, we made the N-gram model from only the newspaper articles by using the conventional method (referred to as the base model). We used the Witten-Bell discounting method [2] for smoothing in both models.

We prepared 100 sentences from the newspaper articles (referred to as NP) and 50 query sentences for the QA system (referred to as QA), and these were recorded for four speakers (two men and two women). Though the word network was relatively small and had only 33 nodes (31 words) in the network, the 36 of 50 query has the frozen patterns written by the network (referred as QA').

The existing N-gram decoder [4] was used for the recognition experiments. The language model weight and the insertion penalty were set to the best values for the newspaper (base) model. The results are shown in Table 1.

We found that the integrated model significantly reduced the word error rate (WER) for the QA queries (QA) while scarcely increased the WER for the newspaper articles (NP). Moreover, when the word network covered the frozen patterns of the queries (QA'), it reduced the WER further.

6. CONCLUSION

Our selective back-off smoothing method was developed to enable the incorporation of grammatical constraints into the conventional N-gram model. The resulting integrated model is compatible with the conventional back-off N-gram model so that it can work as the language model for an existing N-gram decoder. We applied our model to model the frozen patterns used in queries for question answering systems. Results showed that our model significantly reduced the WER for queries for QA systems while scarcely increased the WER for the newspaper articles that had been modeled by the base N-gram before integration.

7. REFERENCES


