Semi-Automatic Language Model Acquisition without Large Corpora

Tomoyosi AKIBA, Katunobu ITOU

Electrotechnical Laboratory, AIST, MITI

1 Introduction

In this paper, we discuss a methodology for the development of a language model for speech recognition, and introduce a semi-automatic method of acquiring a language model, which does not require large corpora.

Statistical language models have gained a reputation as providing the overall performance for speech recognition, and so widely used in speech recognition systems today. The tasks to which statistical language models can be applied are, however, limited, because a large corpus is essential for the building of a statistical model, and the collection of a new corpus is a very costly task in terms of time and effort. Thus, if our aim is to apply speech recognition to various tasks as required, we need a way of developing a new language model for a given task at a reasonable cost.

On the other hand, our new method (fig.1) is structured so that it can attempt to acquire language models from various knowledge resources. Each knowledge resource makes its own contribution to the acquired language model. For example, novice users may specify sequences of words that are and are not sentences. Experts can specify the constituents that makes a sentence, that is, what is often called grammatical knowledge. Most electronic dictionaries available today carry information about words, including part-of-speech, inflection patterns, semantic class, and so on. Of course, a corpus is considered as one of knowledge resources. In addition, we must consider about speech recognition systems; the acquired language model should be used by them.

To integrate information from such a range of knowledge resources, a uniform representation is essential. In section 2, a specific class of attribute grammars is introduced for this purpose.

In section 3, we introduce a semi-automatic method to acquire a Japanese language model for any new task as required. The EDR electronic dictionary [1], an existing electronic dictionary of the Japanese language, and a small set of example sentences which are intended to convey the characteristics of the task, are used instead of a large corpus. Our method is also intended to utilize the knowledge of experts as much as possible.

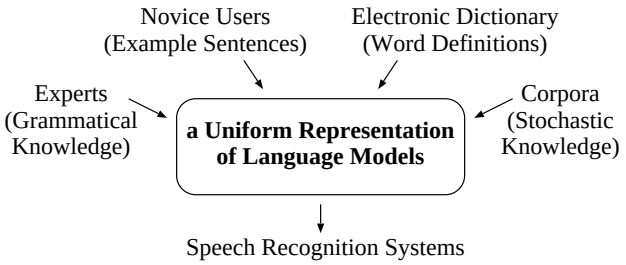


Figure 1: Language modeling from various resources

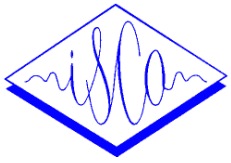
2 The class of attribute grammars

As stated above a uniform representation is indispensable so that various kinds of resources can be used. In particular, a representation that is easy for humans to understand should be used, since the system should be easy for humans to maintain, modify and extend. Most speech recognition systems today use language models based on either regular grammars (RGs) or context-free grammars (CFGs) [1]. But, neither RG nor CFG is easy for humans to understand, because the number of rules required to carry out practical tasks becomes very large.

An attribute grammar (AG) is a natural extension of a CFG. On the basis of the underlying CFG, the symbols are augmented by attributes and the rules are augmented by semantic rules that specify methods to be used to calculate the attribute values. The semantic rule that is selected will vary and this decides the class of AGs. We adopted a comparatively simpler class of AGs as our uniform representation of language models. In the class of AGs we are using, the semantic rules are subject to the following restrictions.

1. The semantic rules can be attached to any symbol in a rule of the underlying CFG. The rules are restricted to be either the assignment of a value to an attribute (the notation is '`<attribute>=<value>`'), or the testing for a condition that the value of an attribute should satisfy (the notation is '`<attribute>^<value>`'). A value is specified as either a constant or an initial value (namely, the value before any rule has been applied) of some other attribute of the same symbol (with the notation '`$<attribute>`').

[1] Most stochastic language models are based on RGs



6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

2. After all semantic rules attached to symbols on the right-hand side of an underlying CFG rule have been applied, the attribute-value pairs of them must be consistent.

3. All consistent attribute-value pairs that are obtained from the right-hand side of a rule are gathered. The semantic rules attached to a symbol on the left hand side is, then, applied on them, obtaining the resulting attribute-value pairs of the symbol on the left-hand side.

Let's look at a example. In the example below, semantic rules are placed in parentheses after symbols of a CFG rule and syntactic sugars with values "!" and "*" are introduced as substitutes for 'no value' and 'some value' (negation of 'no value') respectively.

Suppose a AG rule:

```
VP(X=!) -> PP(sem=!,mod^*=!,X=$mod)
           VP(sem^*,X=$sem)
```

The rule can be applied if the symbols on the right-hand side have the attribute-value pairs:

```
PP = {(sem,3ce7d1)(mod,3cea06)}
VP = {(sem,3cea06)}
```

The resulting attribute-value pairs for the symbol VP on the left-hand side are the same as the pairs for VP on the right-hand side.

On the other hand, the rule cannot be applied if the symbols have the attribute-value pairs:

```
PP = {(sem,3ce7d1)(mod,3cea06)}
VP = {(sem,3cf15e)}
```

because the values of the attribute 'X' on the right-hand side are inconsistent.

Although this class of AGs is a quite simple extension of CFGs, it has great advantages for our purposes. Its application can result in a big reduction in the number of rules required for a given task. For example, in section 4 we introduce a language model for a town guidance task, and its AG only needs 50 rules, though the equivalent CFG would need over 1000. This makes it relatively easy for humans to revise and to improve the model. Nevertheless, this class of AGs is equally descriptive; any AG of this class can be converted into an equivalent CFG. [2] This makes the model easy to use in practical speech recognition systems.

3 The process of language modeling

Our process of language modeling is shown in fig.2. We used a small set of example sentences and an existing Japanese-language electronic dictionary, the EDR electronic dictionary[1], instead of a large corpus.

We divide the information in a language model into two categories; task-independent and task-dependent.

[2]The proof is obvious from our definition because the set of attribute-value pairs of our AG are finite.

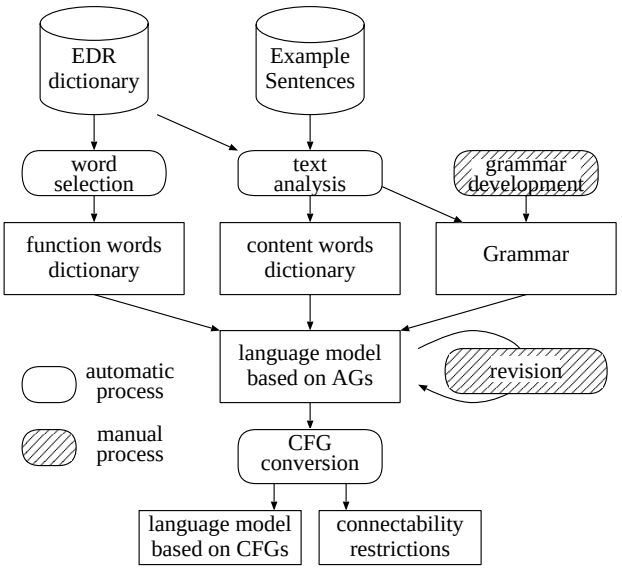


Figure 2: The process of language modeling

Task-independent information includes general knowledge about function words and the syntactic knowledge that provides grammatical sentences. Task-dependent information includes the selection of the content words that are used in the task and the semantic information that provides the possible modification between two words. Basically, Task-independent information is extracted directly from the EDR electronic dictionary, and task-dependent information is extracted from example sentences with the assistance of the EDR dictionary. Note that the body of task-independent information can be applied to any other tasks after once it has been obtained. The task-independent and task-dependent models are then merged. As the models are based on AGs, they can be revised by experts as required. Finally, the (revised) language model based on AGs is converted to the equivalent model based on CFGs so that it is suitable for use by existing speech recognition systems.

In each process, we introduce several levels of approximation, each of which corresponds to a degree to which an expert's knowledge is needed in constructing the model. In the first level, the model is constructed in a fully automatic process. In the later levels, the model is no longer constructed automatically but now only semi-automatically under an expert's supervision. In general, the more knowledge supplied by experts, the better the resulting model.

3.1 The extraction of content words dictionary

The content words dictionary is automatically extracted from example sentences. The EDR Japanese Word Dictionary is used to morphologically analyze the sentences and obtain their content words. Dictio-

nary entries for the words are then automatically constructed. Each entry has attributes that have been extracted from the EDR dictionary. For example, the following entry for the word "station" has four attributes (part-of-speech, connectabilities, and semantic-ID).

```
T(cat=JN1,left=JLN1,right=JRN1,sem=3d0603)
    -> "e k i"
```

In the first level of approximation (used as a part of the language model A and A' in the following experiment), only connectability information, the information on adjacent words to which the given word can be connected, is attributed. This is expressed as a pair: the right-connectable and left-connectable categories that are defined in the EDR dictionary. The possible connections between two categories are also defined by the EDR dictionary. In the second level (model B and B'), part-of-speech information is also attributed to each entry, so that the expert can build a grammar based on the dictionary. In addition, a semantic-ID attribute is added at the third level (model C and C') as a further constraint on the resulting model. The grammatical rules that are used for semantic constraints are also automatically extracted in this level. They are obtained by analyzing the dependency structures of the example sentences. The following rule:

```
T(mod=3cea06) -> T(sem^3ce7d1,mod^!)
```

says that the word with semantic-ID 3ce7d1 ("place") can modify the other word that has semantic-ID 3cea06 ("teach").

3.2 The extraction of function words dictionary

The function words dictionary is extracted directly from the EDR Japanese Word Dictionary. The connectability information and the part-of-speech information are attached as the attributes to each dictionary entry. In the first level of approximation(used as a part of the language model A, B, and C), words are selected automatically only by the frequency of use as defined in the EDR dictionary. In the next level (model A', B' and C'), we inspected and refined the automatically created dictionary. We trimmed the inappropriate entries by hand, because some words appeared to be seldom or never used in conversation, and added some entries that seemed to be important but had not been selected by the automatic process.

3.3 The development of the grammar

Basing a grammar on the extracted dictionary improves the resulting model, because it restricts the possible sequences of words. Of course, we can choose not to use one (that is, to use a dummy grammar that allows all sequences). In the first level of approximation (model A and A'), a simple grammatical rules that allows all word sequences is used (fig.3). These rules only restrict allowed connections with adjacent words according to the connectability attribute of each dictionary entry. In the second level (model B and B'), we included the restriction of syntactically ill-formed sentences in the grammar(fig.4). The part-of-speech attribute of each dictionary entry was used here. In the third level (model C and C'), we included the restriction of sentences to those with semantic structures that appear in the example sentences. The semantic-ID attribute of each dictionary entry was used for this.

```
S -> NT
S -> S NT
NT(left=!,right=!) -> T(left^*,right^*)
```

Figure 3: Simple grammatical rules

```
PP -> NP PROPS
NP -> PP(mv=!,mn^1=!,fn=!) NP
NP -> VP(mv=!,mn^1=!,fn=!) NP
NP -> NOUN
NOUN(left=!,right=!) -> T(cat^JN1)
```

Figure 4: Rules that provide a syntactic constraint

3.4 Handling the resulting model

The content words dictionary, the function words dictionary, and the grammar are all merged into a full language model for a specific task. They all are based on AGs, so experts can easily revise or refine the model at any time. For example, as a content words dictionary which is automatically generated by the morphological analysis is likely to have errors, it might be a good idea to correct them by hand.

The resulting language model based on AGs is converted to a CFG so that it can be used in existing speech recognition systems. Though any AG can be converted into a CFG alone, we have converted it into a CFG and rules for allowed connections between adjacent words. This considerably reduces the number of rules in the CFG. The CFG rules and rules for connection are incorporated in a single LR table[2].

4 An Experiment

We used our method to build a language model for a spoken dialog task for which a corpus of spoken language that was collected by using the WOZ technique is available. The corpus, the "ETL Spoken Dialog Corpus on Town Guidance Task"[3], contains data collected from forty speakers during 197 sessions. The 192 utterances of 5 sessions of 5 speakers were chosen and used as the example sentences for making the language model for this task. Six models (A, A', B, B', C, and C') were made according to the levels of approximation outlined in section 3. The six models, and another model based on CFGs constructed entirely by hand, are investigated in terms of their perplexity and

language model	characteristics # of rules	# of words	perplexity	coverage (on 2943 sentences) # of sentences	%
A (connectability only)	5	1023	181.9	2124	72.2
A' (with function words selection)	5	996	186.4	2128	72.3
B (A + syntactic constraints)	50	979	155.2	1917	65.1
B' (with function words selection)	50	906	147.3	1953	66.4
C (B + semantic constraints)	50+75	979	91.3	1748	59.4
C' (with function words selection)	50+75	906	87.2	1783	60.6
the CFG created by hand	1462	386	68.3	1759	59.8

Table 1: The characteristics and the coverage of the models

coverage onto other 2943 utterances in the corpus. The result of the experiment is shown in table1. It shows that, considering the fact that there always is a trade-off between perplexity and coverage, the models created by using our method are slightly worse than the model made by hand. There was, however, a much smaller cost to build the models of our method. Note further that, they have far fewer rules than the model made by hand; this would makes it relatively easy for humans to improve them.

5 Related Works

Levison et al.[4] have used AGs for natural language generation. Their class of AG is similar to ours. The difference is that their approach was to propagate attributes in a top-down fashion (because their system is used for generation), while in ours attributes propagate in a bottom-up fashion (because our system is for recognition).

The automatic acquisition of grammatical knowledge has been studied mainly in the realm of natural language processing (NLP)[5]. Such approaches differ from ours on three major points. Firstly, most works in NLP is based on the acquisition from corpora alone, while our system handles information from various knowledge resources, in order to overcome the shortage of training corpora. Secondly, in works on NLP, the acquired knowledge is represented based on CFGs and therefore the number of rules becomes huge. On the other hand we have used AGs, which results in considerably fewer rules and make it possible for humans to revise them. Thirdly, the characteristics of grammatical knowledge are quite different in NLP and in speech recognition (SR). In NLP, grammatical knowledge is utilized to give structures to sentences, and therefore the process often does not consider ill-formed inputs. On the other hand, in SR, so in our method, the knowledge is utilized to reject ill-formed inputs.

6 Conclusion

A method for the acquisition of a language model from a variety of knowledge resources is discussed. Instead of large corpora, a small set of example sentences, the knowledge of experts, and a readily available electronic dictionary have been used to build a Japanese language model. The method can be applied to any new task required at a reasonable cost.

Note that, as statistical knowledge can also contribute to improving the language model, there is no reason it should not be used when it is available. There is some work about the integration of grammatical knowledge and statistics[6]. We intend to modify such a model for speech recognition and utilize it for improving the resulting language models.

References

[1] Japan Electronic Dictionary Research Institute, Ltd. EDR Electronic Dictionary Technical Guide. TR-042, 1993.

[2] H. Li and H. Tanaka. A method for integrating the connection constraints into an LR table. In Proceedings of the Natural Language Pacific Rim Symposium (NLPRS95), pp.703-708, 1995.

[3] K. Itou, T. Akiba, O. Hasegawa, S. Hayamizu, and K. Tanaka. A Japanese spontaneous speech corpus collected using automatically inferencing Wizard of OZ system. the Journal of the Acoustical Society of Japan, Vol.20, No.3, May 1999.

[4] M. Levison and G. Lessard. Application of Attribute Grammars to Natural Language Sentence Generation, in "Attribute Grammars and Their Applications", Lecture Notes in Computer Science, Vol.461(1990), Springer-Verlag.

[5] M. Kiyono and J. Tsujii. Hypothesis selection in grammar acquisition. In Proceedings of the 14th International Conference on Computational Linguistics, Vol.2, pp.837-841, 1994.

[6] K. Inui, V. Sornlertlamvanich, H. Tanaka, and T. Tokunaga. A new formalization of probabilistic GLR parsing. In Proceedings of the 5th International Workshop on Parsing Technologies, pp. 123-134, 1997.