

Embedding Grammars into Statistical Language Models

Harald Hüning, Manuel Kirschner, Fritz Class, Andre Berton, Udo Haiber

DaimlerChrysler AG, Dialogue Systems
P.O. Box 2360, 89013 Ulm, Germany
harald.huening@daimlerchrysler.com

Abstract

This work combines grammars and statistical language models for speech recognition together in the same sentence. The grammars are compiled into bigrams with word indices, which serve to distinguish different syntactic positions of the same word. For both the grammatical and statistical parts there is one common interface for obtaining a language model score for bi- or trigrams. With only a small modification to a recogniser prepared for statistical language models, this new model can be applied without using a parser or a finite-state network in the recogniser. Priority is given to the grammar, therefore the combined model is able to disallow certain word transitions. With this combined language model, one or several grammatical phrases can be embedded into longer sentences.

1. Introduction

This work aims at combining the best of two worlds regarding the language models used for speech recognition. On the one hand, statistical language models are employed for the recognition of spontaneous speech with a great variety of expressions, on the other hand, grammars are used for command and control applications. We consider deterministic speech recognition grammars which represent all utterances that are to be recognised. Typically, these command and control grammars are very limited in the ways of expression that the speaker may use. The problem of the applications using such grammars is that the speakers need to learn a set of allowed commands.

While grammars can be written directly, statistical language models are trained from training texts by estimating conditional n-gram probabilities. Here we consider class-based trigram language models with back-off to lower n-grams [1]. Due to the n-gram contexts and back-off, the statistical language models are to some extent prepared to recognise new utterances that are not part of the training text. So, statistical language models have an advantage in recognizing a great variety of utterances compared to grammars. In contrast, the advantage of grammars is a better recognition rate. This work aims at combining the variety of expression from statistical language models with the better recognition rates from grammars in the same sentence.

1.1. Other approaches

Hennecke & Hanrieder [2] suggest to use statistical language models for filler words only (semantically empty words), and a grammar otherwise. It is marked in the grammar at which position in a phrase the filler words may occur. However, it is not described how the two types of language model could work together in a recogniser.

It is desirable to employ the grammar in the recognition process directly, because a grammar can disallow word transitions. In contrast to statistical language models, this rejection of word transitions can lead to better recognition rates.

Some approaches use a context free grammar parser. The parser predicts allowed successor words or filters hypotheses proposed by the statistical language model. For example, a probabilistic Earley parser [3] can be used, or an LR-parser combined with a weight-based grammar/SLM scoring scheme [4]. When such elaborate systems are used in the decoding phase, run-time efficiency becomes an issue.

Grammars require in principle information on the complete history of words from the beginning of the sentence (all predecessors or a word), or some equivalent information like parse stacks [3][4]. For example by storing parse stacks with the word hypotheses (a graph or lattice), one can avoid the problem of tracing back the history of words from the beginning of the sentence. As an alternative to parse stacks, Soltau [5] proposes to store instances of past words in addition to the recognised words, or the most common solution is to store the position in a graph representing the grammar (finite-state network).

Lin et. al. [6] combine class-based statistical language models with grammars represented as finite-state networks. A special class label in the statistical language model branches into the grammar. These authors use empirically set weighting factors to encourage entering a grammar coming from a statistical language model, and to punish leaving the grammar before reaching its internal final state. Generally, when entering a grammar language model state, the decoder must also simultaneously consider an alternative path outside this model (i.e. traversing just the general word-based LM). This alternative path eventually becomes the higher-ranking language model hypothesis if the utterance should deviate massively from the grammar model.

1.2. Syntactical Bigrams and the DC research recogniser

Our DaimlerChrysler research speech recogniser uses bigrams in a first pass of the decoder, and uses trigrams in a second rescoring pass. Instead of n-best lists, we use a word hypothesis graph [7]. In the word graph, the predecessor words required for n-grams are not unique, but usually incur a search back in the graph.

For applying a grammar, more context than n-grams is required. The current method avoids using the history from the beginning of the sentence or using a grammar parser with a stack. The grammar is pre-compiled into word bigrams with word indices. There is additional overhead, because words are duplicated with different indices. The different word indices are typically obtained by indexing a determined and minimised graph of the grammar, and they serve to represent

syntactic constraints locally. The bigrams with word indices are also called syntactical bigrams [8]. In the following example the word ‘radio’ has two different syntactic positions. Therefore the indices `_0` and `_1` are used:

```
Radio_0 Charivari_0
Hit_0 Radio_1
```

The word indices serve to decide on the basis of bigrams only that sentences like “Hit Radio Charivari” are not allowed. The word indices enable a recogniser to work with bigrams only, because a single predecessor word is enough to identify the history from the beginning of the sentence.

When the syntactical bigram grammar is applied in our recogniser, the value for a grammatical transition is chosen as 0.01 (highest) and disallowed transitions are assigned a very small number like the lowest floating point value. Using syntactical bigram grammars is feasible as long as we consider command grammars, and do not attempt to model whole sentences with this grammar alone.

The next chapter presents the method to combine statistical language models with grammars, based on the representation of a grammar by means of word bigrams with word indices. Then we present some preliminary experimental results and discuss our results.

2. Combining statistical language models with grammars

This method aims at combining the high recognition rates for those parts of sentences that are matching to a grammar with the recognition of spontaneous speech. Furthermore, the recogniser should be able to switch between grammatical and statistical word transitions within one sentence in an alternating way. Here the above described grammar mechanism of word bigrams with word indices [8] is combined with a statistical language model. The grammar represented as syntactical bigrams can be used by a speech recogniser in a fashion very similar to statistical n-gram language models. Drawing on this similarity, a method to combine syntactical bigram grammars with statistical language models has been developed [9].

The word indices save the need for any other mechanisms than n-gram language models to implement the grammar for a speech recogniser. Let us at first consider to have the word indices in the HMM lexicon, giving more words in the lexicon than otherwise. An alternative without overhead in the lexicon is considered in [9], but for illustration it is easiest to consider the word indices coming from the lexicon.

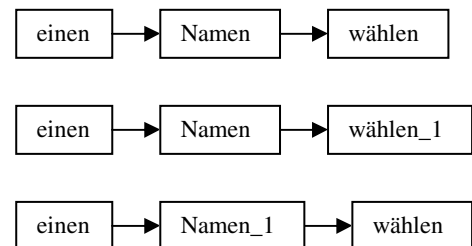
2.1. A common interface

Assume that the HMM lexicon contains both words with and without word indices. Then the decision about using the statistical model or the grammar to assess a word transition can be carried out merely based on the presence of word indices at adjacent words (bigrams). In this way, the grammar and the statistical language model have a common interface, they can be regarded as two parts of one language model. Only word bigrams or trigrams need to be given, and in response, language model probabilities between 0 and 1 are returned.

The presence of word indices allows a distinction about which word transitions are belonging to the grammar and which ones are not. Therefore we now also use the expression grammar words for those words that have word indices.

Figure 1 illustrates the distinction of which n-grams are assessed by the grammar or statistical part. When the recogniser asks the combined language model for a trigram probability, it is first checked if the last two words (bigram) contain word indices. If this is the case, the word transition belongs to the grammar, and the language model only needs a bigram (no trigram). In all other cases statistical probabilities (trigram or back-off) are used. Note that the markers for the beginning and end of a sentence are also considered as grammar words.

a) mixed or plain statistical n-grams



b) bigram to be assessed by the grammar only

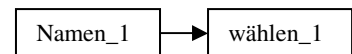


Figure 1 The presence of word indices determines whether given bi- and trigrams are treated by the grammar or statistical part of the language model. a) mixed n-grams or n-grams without indices will be assessed by the statistical language model, b) when both of the last two words have word indices, the grammar has priority over the statistical language model.

In contrast to usual statistical language models, this combined method can return the information that a transition is not allowed. The decision by the grammar that a word is allowed or not allowed is also conveyed by means of probability values. For the common interface the grammar uses a high and a low probability value. The low value may be chosen near the lowest floating point number to represent a forbidden word transition. For the high value it is convenient to choose the value of 1, so no statistical transition can be assessed better than the grammatical transitions. Only the word penalty of the statistical part (a factor per word, see [7]) remains present in the language model score for the grammar as well. We have found that a word penalty of 0.01 is often optimal for the statistical language model, so this is in good accord with the chosen high value of 0.01 of the grammar, see section 1.2.

The language model score for a hypothesised sentence consists of a product of bi- or trigram (or back-off) probabilities together with the word penalties depending on the number of words, and special factors for pauses. Through the common interface of our syntactical bigram grammar and the statistical language model, both the decoding and word graph (or lattice) rescoring methods can be used just like for standard statistical language models.

Assuming the word indices are stored in the lexicon, it is straightforward how the word hypothesis graph with word

indices is produced by the decoder. The decoder places copies of some words with different indices as parallel paths into the graph. The different paths serve for comparing the different syntactic alternatives. When the decoding is completed and the graph covers the whole sentence, some paths will not be continued until the end of the sentence. For example, those paths will be pruned where the grammar does not allow any ongoing transition.

A special case of language model probabilities may be implemented for the treatment of garbage words (hesitations, pause, noise). For example, one may want to allow garbages before and after every word of a grammar. This can be achieved by ignoring any garbages for the language model history. However, the recognition of garbages may still require the language model to return probabilities for garbage words.

2.2. Transitions within one sentence

As Figure 1 has shown, next to the purely statistical trigrams and the bigrams with two word indices, there can be mixed trigrams. The mixed trigrams can have word indices in some positions, but not at both of the last words. They are treated by the statistical part of the language model, and with their help there can be any number of transitions between the two models, grammatical and statistical n-grams, in the same sentence, see Figure 2. In the Figure, the transition “Namen_1 wählen_1” is treated by the grammar, all other n-grams are assessed by the statistical model. Embedding the grammar phrase into the more natural sentence furthermore requires that in this example the following mixed n-grams are trained sufficiently: “möchte einen Namen_1” and “Namen_1 wählen_1 bitte”. Inserting words into the middle of grammatical phrases is not allowed. So the topmost path in the Figure “einen Namen jetzt wählen bitte” is only allowed by the statistical model. The combined model is flexible about alternating its two parts at any position in a sentence. The mixed n-grams can be found in the first place by including word indices in the training text for the statistical

language model. The addition of word indices can be made by conducting a search for the grammar phrases in a copy of the training text. Where a grammatical phrase is identified in sentences of the training text, the particular word indices of the words in this phrase are added to the text. In this way, words in the training text are tagged with word indices depending on their context. For class-based statistical language models, a mapping from words to classes may have to be observed when identifying phrases in the training text. Furthermore, two grammar phrases may follow each other directly. If this occurs in the text, the effect in the trained model is nearly identical to a longer phrase in the grammar. After tagging the grammatical phrases in the text, n-grams are counted, and so the mixed n-grams are found.

The mixed n-grams are only an addition to the original statistical language model, not replacing the original n-grams. Their probabilities are copied from the n-grams without word indices rather than calculating lower probability values due to the larger number of successor words with word indices. The n-grams that differ only in their word indices can share the same probability value, because the syntactical information is only important for the grammar and should not change the probabilities. Therefore, strictly speaking the sum of conditional probabilities in such language models is often greater than 1.

The final language model is created by merging three parts: the n-grams for transitions (mixed n-grams) as well as the grammatical bigrams and the original statistical language model. No trigrams are required for the grammar, assuming a small modification is made to a recogniser for statistical language models: if the language model software implements the decision to look for bigrams only when the last two words have indices, then no trigrams are required for the grammar.

3. Experimental results

We have performed preliminary tests of the combined language model with dialogs recorded in cars. The aim of a Wizard-of-Oz setup was to let the subjects speak naturally,

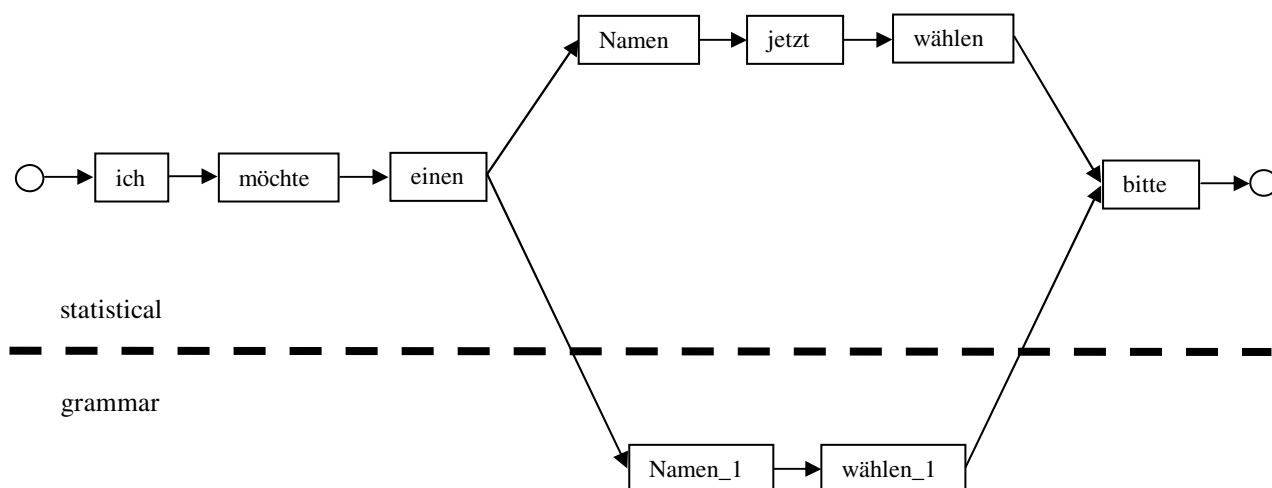


Figure 2 Transitions between statistical and grammatical n-grams for the sentence “ich möchte einen Namen (jetzt) wählen bitte”. In this example, the insertion of the hypothesis “jetzt” is not allowed by the grammar.

only afterwards a grammar has been developed to cover the most commonly used phrases. We have tested 1810 sentences (5554 words) on telephone and address book applications in German language. More details of the recogniser and the tests will have to be presented later, here we show the variation of a weighting parameter between the grammar and the statistical part. We have found that the statistical part should have a greater weight than the grammar, so a factor smaller than 1 is applied to the value of each grammatical bigram. Figure 3 shows the word error rate for different values of this weight. An optimum for this test set is roughly at 0.3.

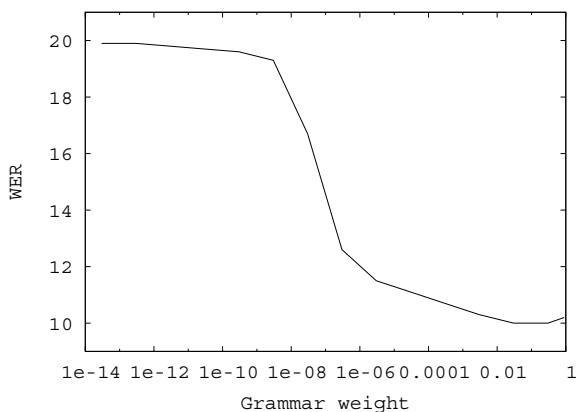


Figure 3 Word error rate for different weighting factors between the grammar and statistical part of the language model. The optimum is at about 0.3.

4. Discussion

In our combined model the grammar and statistical model can alternate arbitrarily within one sentence. Thus several grammar phrases can be recognised in one sentence, as in the work of Lin et al. [6]. Instead of inserting markers for the grammar into a statistical language model, we introduce specific n-grams for the transitions into grammar phrases. We use a weighting factor similar to their boost factor [6], however, our factor is applied at every n-gram. We do not yet know if this value needs to be optimised for every new application, in which case an adaptation technique might be useful.

In contrast to Lin et al. [6] we do not allow early exiting from a grammatical phrase. If only the beginning of a phrase is hypothesised, then the whole path matching the grammar may be pruned away.

An open issue is the behaviour of confidence measures with the additional parallel paths in our word graph. We do not discriminate between the word indices in calculating a graph homogeneity measure. So the words with many indices always have an advantage in the confidence measure. We have not yet experimentally compared the confidence values with and without the combined language model.

5. Conclusions

We have combined syntactical bigrams with statistical language models. Only a small change is required to a speech recogniser that uses statistical language models and graph rescoring: the decision module that avoids back-off for transitions between grammar words. Priority is given to the

grammar, which supports more reliable recognition of grammatical phrases. A limitation of the method is that the word indices can only be handled for small grammars, typically phrases shorter than 5 words are advisable. However, this may be enough for phrases to embed into statistical language models.

Our first experiments on in-car dialogs indicate that the recognition performance is optimum for the combination with a slightly stronger weight on the statistical language model.

6. Acknowledgements

This work was partly funded by the German Ministry of Education and Research (BMBF) in the framework of the SmartWeb project under grant 01 IMD01 D. The responsibility for the content lies with the authors.

7. References

- [1] Clarkson, P. R. and Rosenfeld, R., "Statistical Language Modeling Using the CMU-Cambridge Toolkit", *Proceedings ESCA Eurospeech*, 1997.
- [2] Henneke, M. E. and Hanrieder, G., "Easy Configuration of Natural Language Understanding Systems", *Proceedings of Voice Operated Telecom Services, Do they have a bright future?*, COST 249, Ghent, Belgium, 2000.
- [3] Jurafsky, D., Wooters, C., Tajchman, G., Segal, J., Stolcke, A., Fosler, E. and Morgan, N., "Using a Stochastic Context-Free Grammar as a Language Model for Speech Recognition", *Proceedings of ICASSP-95*, pages 189-192, Detroit, MI, 1995.
- [4] Wachsmuth, S., Fink, G. and Sagerer, G., "Integration of parsing and incremental speech recognition", *Proceedings of the European Signal Processing Conference*, volume 1, pages 371-375, Rhodes, Sep. 1998.
- [5] H. Soltau, F. Metzger, C. Fügen & A. Waibel. "A one pass decoder based on polymorphic linguistic context assignment." In *Proceedings of the Automatic Speech and Recognition Workshop (ASRU)*, Trento, Italien, 2001.
- [6] Lin, Q., Lubensky, D., Picheny, M. and Srinivasa Rao, P., "Key-phrase spotting using an integrated language model of n-grams and finite-state grammar". *Proc. Eurospeech-97*, pp. 255-258, 1997.
- [7] Kuhn, T., Fetter, P., Kaltenmeier, A. and Regel-Brietzmann, P., "DP-Based Wordgraph Pruning", *Proceedings ICASSP'96*, volume 2, pp. 861, Atlanta, USA, 1996.
- [8] Kilian, U., Class, F., Kaltenmeier, A. and Regel-Brietzmann, P., "Representation of a Finite State Grammar as a Bigram Language Model for Continuous Speech Recognition." In *Proc. European Conf. On Speech Communication and Technology*, pp. 1241-1244, Madrid, Sept. 1995.
- [9] Berton, A., Class, F., Haiber, U. and Hüning, H., "Verfahren zur Spracherkennung von Wortfolgen", Offenlegungsschrift (patent application) DE 103 35 569 A1, 2003.