Language Model Adaptation Using Different Class-Based Models

Aleš Pražák, Pavel Ircing and Luděk Müller
University of West Bohemia, Faculty of Applied Sciences, Department of Cybernetics
Plzeň, Czech Republic

Abstract

The paper presents two different methods for adding previously unseen words into the LVCSR system. Both methods employ the principles of class-based language modeling – the first one exploits task-specific knowledge, the second one is fully automatic and task independent. Extensive test of the proposed language models in the real-time ASR system showed that both techniques provide a consistent improvement in terms of recognition accuracy. Moreover, the contributions from both methods appear to be additive, yielding a total improvement of up to 2 % absolute.

1. Introduction

The language model (LM) constitutes one of the key parts of the automatic speech recognition (ASR) system, as it provides useful constraints for the acoustic analysis (only words appearing in the LM can be recognized by the decoder) and helps to resolve many ambiguities between similarly sounding words using information about the word context. On the other hand, the restrictive function of the language model becomes a burden when we are trying to recognize words that did not appear in the language model training data and thus are not present in the LM. In such a case, the problem is actually two-fold – first, we have to identify which words that are currently missing in the LM are going to be needed for a successful recognition and second, we have to find a method of their incorporation so that context information embedded in the existing LM can be exploited. This paper focuses mostly on the second problem; however, one of the presented approaches tackles also the first issue. All proposed techniques are thoroughly tested on the task of on-line transcription of the broadcasts from the Chamber of Deputies of the Czech Parliament.

2. Adaptation Techniques

We present two methods for adding new words to a language model. Both techniques use the well-known framework of class-based language modeling but each of them in a slightly different way. The first one is based primarily on the specific knowledge about the task in question, whereas the second is general and completely task-independent.

2.1. Name Classes

This method stemmed from the following, relatively simple considerations. First, the names (both personal and geographical) often constitute an Out-of-Vocabulary (OOV) words, since they are usually underrepresented in the training data (that is, if we put aside the notorious ones such as celebrity names or capital cities). Moreover, especially the rare names are very important if we want to use the automatically transcribed text for information retrieval purposes, since people frequently search for names or named entities in general. On the other hand, for domain-specific ASR tasks it is often feasible to obtain a list of names that are to be expected in the utterances – for example, line-ups for the recognition of the team-sport broadcasts or, as in our case, the complete list of parliament members and government ministers. The names from these lists can then be put into the manually designed classes and consequently exploit the context information present in the class-based language model even in the case where they did not appear in the original LM training data.

Because of the highly inflectional nature of the Czech language, 5 different classes were designed for personal names:

1. Last_name or First_name_last_name combination in either nominative or vocative
2. Last_name or First_name_last_name combination in either genitive or accusative
3. Last_name or First_name_last_name combination in either dative or locative
4. Last_name or First_name_last_name combination in instrumental
5. Last_name_first_name combination

The names of all parliament members and ministers appearing in training text were replaced with the corresponding class tags and a complete list of the politicians of a given electoral period is automatically inflected and added to the appropriate classes. Each word can belong to one class only – homographs are artificially distinguished to ensure one-to-one mapping. All individual names within a class have the same probability, but the ratio between Last_name and First_name_last_name combination is computed according to their frequency in the training data.

The resulting language model is a hybrid of a word-based and a class-based language model – the probability of politicians' names is computed according to the class model formula

$$P(w_i \mid h_i) = P(w_i \mid c_i) \cdot P(c_i \mid x_{i-n+1}^{i-1}) \qquad x_i = \begin{cases} c_i & \text{if token i is a class} \\ w_i & \text{if token i is a word} \end{cases} \tag{1}$$

i.e. the *n*-gram probability of a word w_i given the history $h_i = w_{i-n+1}^{i-1} = w_{i-n+1} \ldots w_{i-2} w_{i-1}$ is a product of the word probability within the class $P(w_i \mid c_i)$ and the *n*-gram probability that the class c_i will follow the previous $(n-1)$ tokens (either words or classes) - $P(c_i \mid x_{i-n+1}^{i-1})$, whereas the probability of other words is determined by a standard word-based *n*-gram formula

$$P(w_i \mid h_i) = P(w_i \mid x_{i-n+1}^{i-1}) \tag{2}$$

2.2. Morphological Classes

The second type of employed language models employs fully automatic word-to-class mapping and does not require any task-specific knowledge. In this case, all words are mapped to (possibly multiple) classes based on morphological tag from tagset presented in [1]. Every tag in this tagset is represented as a string of 15 symbols. Each position in the string corresponds to one morphological category in the following order - part of speech, detailed part of speech, gender, number, case, possessor's gender, possessor's number, person, tense, degree of comparison, negation and voice. Positions 13 and 14 are currently unused and finally position 15 is used for various special purposes (such as marking colloquial and archaic words or abbreviations). Non-applicable values are denoted by a single hyphen (-). For example, the tag `VB-S---3P-AA---` denotes the verb (`V`) in either the present or the future tense (`B`), singular (`S`), in the third person (`3`), in the present tense (`P`), affirmative (`A`) and in the active voice (`A`). The morphological tagging is performed automatically using a serial combination of the Czech morphological analyzer and tagger [2].

The n-gram probability of a word w_i given the history h_i is in this case given by

$$P(w_i \mid h_i) = \sum_{c_{i-n+1} \in C_{i-n+1}} \ldots \sum_{c_{i-1} \in C_{i-1}} \sum_{c_i \in C_i} P(w_i \mid c_i) \cdot P(c_i \mid c_{i-n+1}^{i-1}) \qquad C_i = \{c : w_i \in c\} \tag{3}$$

This formula is different from (1) because all words are mapped into classes this time and the word-to-class mapping is one-to-many (that is where the sums come from).

It is generally known (and our previous experiments have proved it [3]) that the "full" class-based model yields more robust probability estimates than the word-based one but at the same time it has worse discrimination ability. Therefore word-based and class-based models are usually combined in some way. We have found out empirically that the following exponential (log-linear) combination

$$P(w_i \mid h_i) = P_{LMC}(w_i \mid h_i)^{\alpha} \cdot P_{LMW}(w_i \mid h_i)^{(1-\alpha)} \tag{4}$$

of the class-based (LMC) and word-based (LMW) language models with weighing factor $0 \le \alpha \le 1$ gets better results than a simple linear combination

$$P(w_i \mid h_i) = \alpha \cdot P_{LMC}(w_i \mid h_i) + (1-\alpha) \cdot P_{LMW}(w_i \mid h_i). \quad (5)$$

3. ASR System Description

The fast 2-pass large vocabulary continuous speech recognition (LVCSR) system working in real-time developed at the Department of Cybernetics; University of West Bohemia was employed for experiments.

3.1. Acoustic Processing

The analogue input speech signal is digitized at 44.1 kHz sampling rate and 16-bit resolution format. The aim of the front-end processor is to convert continuous acoustic signal into a sequence of feature vectors. We performed experiments with MFCC and PLP parameterizations. The best results were achieved using 27 filters and 12 PLP cepstral coefficients with both delta and delta-delta sub-features. Feature vectors are computed at the rate of 100 frames per second.

Each individual basic speech unit is represented by a three-state HMM with a continuous output probability density function assigned to each state. In this task, we use only 8 mixtures of multivariate Gaussians for each state. The choice of an appropriate basic speech unit with respect to the recognition network structure and its decoding is discussed later.

3.2. Recognition Network

Our LVCSR system uses a lexical tree (phonetic prefix tree) structure for representation of acoustic baseforms of all words of the system vocabulary. In a lexical tree, identical initial substrings of word phonetic transcriptions are shared. This can dramatically reduce the search space for a large vocabulary, especially for inflectional languages, such as Czech, with many words having the same word stem. The automatic phonetic transcription (with pronunciation exceptions defined separately) is applied to all words of the system vocabulary and resulted word baseforms for all pronunciation variants are added to the lexical tree.

In order to ensure better modeling of the word pronunciations, we used triphones (context-dependent phonemes) as the basic speech units. By using a triphone lexical tree structure, the in-word triphone context can be easily implemented in the lexical tree. However, the full triphone cross-word context leads to fan-out implementation by generation of all cross-word context triphones for all tree leaves. This results in enormous memory requirements and vast computational demands. To respect the requirement of the real-time operation we have proposed an approximation of the triphone cross-word context.

One of the possible approaches is to use monophones (context independent phonemes) instead of triphones at the word boundaries. However, this brings the necessity to train two different types of acoustic model units and also mutually normalize the monophone and triphone likelihoods. To cope with this problem, we use only triphone state likelihoods and merge the triphone states corresponding to the same monophone within a given phone context. As the system vocabulary is limited, not all right and left cross-word contexts have to be modeled. This approach results in so-called biphones that represent merged triphone states with only one given context - right in the root and left in the leaves. The biphone likelihood is computed as the mean of the likelihoods of merged triphone states. The proposed biphone cross-word context represents a better approximation than a simple replacement of triphones by monophones on the word boundaries. In addition, this approach increases neither the recognition network complexity nor the decoding time, but only the duration of off-line recognition network creation.

3.3. Recognition Network Decoder

Since bigram language model is implemented in the first pass, a copy of a lexical tree for each predecessor word is required. The lexical tree decoder uses a time-synchronous Viterbi search with

token passing and effective beam pruning techniques applied to re-entrant copies of a lexical tree. The beam pruning is used inside and also at the level of the lexical tree copies, but a sudden increase of hypothesis log-likelihoods occurs due to application of language model probabilities at time of word to word (lexical tree to lexical tree) transitions. Fortunately, early application of the knowledge of language model can be carried out by factorizing language model probabilities along the lexical tree. In the lexical tree, more words share the same initial part of their phonetic transcriptions and thus only the maximum of their language model probabilities is implemented towards the root of the lexical tree during the factorization. In addition, commonly used linear transformation of language model log-likelihoods is carried out for optimal weighting of language and acoustic models.

To deal with requirement of real-time operation, an effective method for managing lexical tree copies is implemented. The algorithm controls lexical tree to lexical tree transitions and lexical tree copies creation/discarding. The number of lexical tree copies decoded in real-time is limited, so the control algorithm keeps only the most perspective hypotheses and avoids their undesirable alternations, which protects the decoding process from time consuming creation of lexical tree copies. The algorithm also manages and records tokens passed among lexical tree copies in order to identify the best path at the end of the decoding. In addition, for word graph generation not only the best, but several (n-best) word to word transitions are stored.

In the second pass of the recognition system, the word graph is rescored with arbitrary n-gram language model. To allow real-time operation, the word graph creation and its rescoring can be performed even several times per second. The whole LVCSR system can effectively use multi-core computer systems, so the proposed fast 2-pass LVCSR system implementation handles tasks up to 200 000 words in real-time with the delay of on-line transcription about one second.

4. Experimental Evaluation

The experiments with different language models were performed on the task of on-line transcription of the broadcasts from the Chamber of Deputies of the Czech Parliament.

4.1. Experimental Setup

The acoustic model was trained on 40 hours of parliament speech records with manual transcription. We used 42 Czech phonemes. As the number of Czech triphones is too large, phonetic decision trees were used to tie their states. Now, transcription of the broadcasts from the Czech Parliament works with 3 729 different HMM states of a speaker and gender independent acoustic model.

About 18M tokens of normalized Czech Parliament meeting transcriptions were used for language model training using the SRI Language Modeling Toolkit [4].

Two test sets from different electoral periods were prepared to evaluate the influence of different language models and different methods for adding new words to them. The first set denoted as “2002” is from the same electoral period whose data are included in the language model training data; however the actual transcriptions of this test set were excluded from training data. The second test set “2006” is from the next electoral period and no transcriptions from that period are present in the training data. Each test set consists of five parliament speech records, half an hour each, different in their contents. Test set “2002” includes 14 941 words, test set “2006” 15 525 words.

4.2. Experimental Description

The baseline language models are represented by standard word-based 2-gram (for the first pass) and word-based 4-gram (for the second pass) back-off language models with Good-Turing discounting estimated directly using the training data. For exponential combination during word graph rescoring, the class-based 4-gram language model with morphological classes was trained according to the method described in section 2.2. The number of unique morphological tags (thus

the number of morphological classes) is 1 481. These language models involve 112 227 words. Baseline language models are referenced as "TAGS" test.

Next, the inflected names of parliament members and government ministers from corresponding electoral periods that were missing in the language model were incorporated. These names were added to the word-based language models with the residual probability given by language model discounting. The morphological tags of the names (determined without any context) were used for adding to the class-based language model. Baseline language models with added names are referenced as "TAGSPLUS" test.

To compare two proposed methods for incorporation of new words into the language model, inflected names of parliament members and ministers from corresponding electoral periods were added to the language models as described in section 2.1. Hybrid word-based 2-gram and 4-gram language models with name classes were trained. The class-based 4-gram language model was trained considering the name classes too, so the names were withdrawn from automatically derived classes. Language models with name classes are referenced as "TAGSNAME" test.

Finally, incorporation of words that do not have any additional information associated with them (i.e., they are not for example named entities) was checked out by adding the OOV words (except slips of the tongue and unintelligible words) from test sets to the language models. These words were added to the hybrid word-based models from TAGSNAME test with residual probability given by language model discounting multiplied by their frequency in test set. In real applications, incorporated words will be extracted from some external texts with reference to the recognition domain, for example Internet news. Some words occur more frequently then others, so their influence in the language model should be emphasized. Hence the multiplication by word frequency. Word frequency was used also for weights of added word within classes in the class-based 4-gram language model. These models are referenced as "TAGSNAMEOOV" test.

To make on-line transcription of the broadcasts from the Czech Parliament possible, language model probabilities have to be evaluated extremely fast. Unfortunately, class-based language models with one-to-many word-to-class mapping do not meet this requirement. That is why only word-based or hybrid word-based language models with one-to-one mapping are used for the first recognition pass and class-based language models are used only for word graph rescoring in the second pass. The OOV word rate and recognition accuracy with 2-gram language models from the first pass are shown in tables 1 and 2.

Table 1. First pass recognition results for test set "2002"

	TAGS	**TAGSPLUS**	**TAGSNAME**	**TAGSNAME OOV**
OOV word rate	1.48 %	1.47 %	1.47 %	0.78 %
Recognition accuracy	84.49 %	84.49 %	84.39 %	85.09 %

Table 2. First pass recognition results for test set "2006"

	TAGS	**TAGSPLUS**	**TAGSNAME**	**TAGSNAME OOV**
OOV word rate	1.53 %	1.34 %	1.34 %	0.59 %
Recognition accuracy	78.33 %	78.40 %	78.89 %	79.49 %

For test set "2002" 124 names were added with 1 occurrence in test data only. This brings no improvement for TAGSPLUS test and even slight deterioration for TAGSNAME test. This is caused by splitting context information embedded in the baseline language model among all names uniformly without considering their prior probability. For test set "2006", where language model training data do not include any texts from corresponding electoral period, the benefit of name classes is clearly visible.

Tables 3 and 4 show recognition accuracy from the second pass of the recognition system with word-based or hybrid word-based 4-gram language models (weighing factor 0.00), class-based 4-

gram language models (weighing factor 1.00) and their exponential combinations (weighing factors 0.25, 0.50, 0.75).

Table 3. Second pass recognition results for test set "2002"

Weighing factor	TAGS	TAGSPLUS	TAGSNAME	TAGSNAME OOV
0.00	85.51 %	85.51 %	85.44 %	86.15 %
0.25	85.54 %	85.54 %	85.48 %	86.29 %
0.50	***86.63 %***	***86.63 %***	***86.55 %***	***87.34 %***
0.75	86.39 %	86.39 %	86.31 %	87.08 %
1.00	82.83 %	82.83 %	83.07 %	83.90 %

Table 4. Second pass recognition results for test set "2006"

Weighing factor	TAGS	TAGSPLUS	TAGSNAME	TAGSNAME OOV
0.00	79.26 %	79.34 %	79.97 %	80.66 %
0.25	79.29 %	79.41 %	79.94 %	80.70 %
0.50	***80.86 %***	***80.97 %***	***81.41 %***	***82.14 %***
0.75	80.47 %	80.57 %	81.17 %	81.87 %
1.00	76.15 %	76.29 %	76.87 %	77.65 %

5. Conclusion

Two methods for incorporation of new words to the language model were investigated. According to our experiments, if some task-specific knowledge about added words is available, this knowledge should be used to properly exploit the word-context information that is embedded in a language model. When no further information about added words is available, these words can be simply added to the language model with residual probability. Although added words are not present in the best recognition hypothesis in this case, class-based language model significantly improves recognition accuracy by word graph rescoring. In our future work, we want to evaluate recognition results using combined class-based language models directly in the first pass, although this cannot be used for on-line transcription at present.

6. Acknowledgements

This work was supported by the Ministry of Education of the Czech Republic under project MŠMT LC536. The access to the METACentrum computing facilities provided under the research intent MSM6383917201 is appreciated.

References

1. *J. Hajič*: Disambiguation of Rich Inflection (Computational Morphology of Czech). Karolinum, Prague, 2004.
2. *J. Hajič, B. Hladká*: Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. In: Proceedings of COLING-ACL conference, Montreal, Canada, 1998.
3. *P. Ircing*: Large Vocabulary Continuous Speech Recognition of Highly Inflectional Language (Czech). Ph.D. thesis, University of West Bohemia, Pilsen, 2003.
4. *A. Stolcke*: SRILM - An Extensible Language Modeling Toolkit. In: Proceedings of ICSLP conference, Denver, USA, 2002.