

# Fitting Class-Based Language Models into Weighted Finite-State Transducer Framework

*Pavel Ircing and Josef Psutka*

University of West Bohemia, Department of Cybernetics  
Univerzitní 8, Plzeň, 306 14, Czech Republic  
{ircing,psutka}@kky.zcu.cz

## Abstract

In our paper we propose a general way of incorporating class-based language models with many-to-many word-to-class mapping into the finite-state transducer (FST) framework. Since class-based models alone usually do not improve the recognition accuracy, we also present a method for an efficient language model combination.

An example of a word-to-class mapping based on morphological tags is also given. Several word-based and tag-based language models are tested in the task of transcribing Czech broadcast news. Results show that class-based models help to achieve a moderate improvement in recognition accuracy.

## 1. Introduction

The work presented in this paper is motivated by the extremely difficult task of large vocabulary continuous speech recognition of the Czech language. Czech, along with Russian, Polish, Slovak, etc., belongs to the family of Slavic languages. All those languages are highly inflectional, which poses a challenge especially for language modeling, mainly because of the following problems:

### 1. Rapid vocabulary growth

The vocabulary size grows very rapidly with the size of the training corpus. This problem is caused by the aforementioned high degree of inflection (potentially up to 300 / 20 / 200 words (a word is defined by its spelling) for a single verb / noun / adjective lemma, but with frequent cases of systematic homography) and also by a high degree of derivation (use of prefixes and suffixes). The result of this phenomenon is the fact that a Czech vocabulary exceeding 600k words still covers only about 98.5% of tokens in running text whereas in English the 99% coverage can be achieved by a vocabulary containing 50k words only.

Consequently, there is a need for a decoder that can efficiently handle large vocabularies. A good choice is the decoder developed by AT&T Labs-Research [1]. It is built on the basis of weighted finite-state transducers (FST). This framework offers time and space efficiency and also allows a uniform representation of different information sources that are used in automatic speech recognition. However, it is still necessary to restrict the decoder vocabulary to approximately 60k items due to the memory limitations.

### 2. High perplexity of word-based $n$ -gram language models

This fact is usually attributed to the free word order that is also characteristic of highly inflectional languages. However, experiments with a trigram model with permutations [2] showed that the free word ordering does not really pose such a serious problem, especially in the short-term dependencies represented by

the  $n$ -gram language models.

We suspect that the high perplexity is again closely connected with the highly inflectional nature of the Czech language. The idea is as follows - even though available Czech text corpora already reached the size that would be sufficient for training a decent language model for English, the parameter estimates for Czech still remain unreliable due to the higher number of distinct words and consequent high number of the  $n$ -gram language model parameters.

Therefore it is appropriate to use some kind of a class-based language model which reduces the size of the language model parameter space and thus ensures more robust probability estimates given the same amount of training data.

It results from the previous paragraphs that we need to devise a way of incorporating class-based models into the FST framework. The description of such techniques is the main goal of this article. A manner of representing a class-based model as a finite-state automaton has already been introduced for example in [3] but it dealt with many-to-one word-to-class mapping only. We will show that many-to-many word-to-class mapping can be also consistently represented within the finite-state machine paradigm.

The paper is organized as follows - Section 2 describes the basic principles of FSTs used in the AT&T decoder. Section 3 shows the class-based models from the finite-state transducer point of view. Section 4 introduces an example of the class-based model where words are clustered according to their part of speech and finally Section 5 presents an experimental evaluation of the proposed techniques.

## 2. Weighted finite-state machines in speech recognition

The definition of weighted finite state machines depends on the algebraic structure called semiring,  $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$  [4]. A semiring is a set  $\mathbb{K}$  with two binary operations, collection  $(\oplus)$  and extension  $(\otimes)$ , such that  $\oplus$  is associative and commutative with identity  $\bar{0}$ ,  $\otimes$  is associative with identity  $\bar{1}$  and  $\otimes$  distributes over  $\oplus$ .

For example,  $(\mathbb{R}, +, \cdot, 0, 1)$  is a semiring. Since in speech recognition we use negative natural log probabilities and the Viterbi approximation, the proper semiring is defined by  $(\mathbb{R}_+ \cup \{\infty\}, \min, +, \infty, 0)$ . This structure is called the tropical semiring.

The most general finite-state machine, a weighted finite-state transducer [1] over a given semiring is an 8-tuple

$$T = (Q, Q_0, F, \Sigma, \Delta, E, \gamma, \rho) \quad (1)$$

where  $Q$  is the finite set of states,  $Q_0 \subseteq Q$  is the set of initial

states,  $F \subseteq Q$  is the set of final states,  $\Sigma$  is the input alphabet,  $\Delta$  is the output alphabet,  $E \subseteq Q \times \Sigma \times \Delta \times \mathbb{K} \times Q$  is the finite set of transitions,  $\gamma$  is the initial weight function mapping  $Q_0 \rightarrow \mathbb{K}$  and  $\rho$  is the final weight function mapping  $F \rightarrow \mathbb{K}$ .

A transition  $t = (q, a, b, c, r) \in E$  can be viewed as an arc from the source state  $q$  to the destination state  $r$ , labeled with the input symbol  $a$ , the output symbol  $b$  and the weight  $c$ .

A path  $\pi$  in  $T$  is a set of consecutive transitions from  $q$  to  $q'$ , that is

$$\pi = ((q_1, a_1, b_1, c_1, r_1), \dots, (q_n, a_n, b_n, c_n, r_n)) \quad (2)$$

where  $q_1 = q$ ,  $r_n = q'$  and  $r_i = q_{i+1}$  for  $i = 1, \dots, n-1$ . A successful path  $\pi'$  is a path from an initial state to a final state. The input label  $\iota$  of the path  $\pi'$  is the concatenation of the labels of its constituent transitions, i.e.

$$\iota(\pi') = a_1 a_2 \dots a_n \quad (3)$$

and analogically the output label  $o$  of the path  $\pi'$  is defined as

$$o(\pi') = b_1 b_2 \dots b_n \quad (4)$$

The weight  $\omega$  associated to  $\pi'$  is the  $\otimes$ -product of the initial weight function value for a given initial state  $q_1$ , the weights of its constituent transitions and the final weight function value for a given final state  $r_n$ , that is

$$\omega(\pi') = \gamma(q_1) \otimes c_1 \otimes c_2 \dots \otimes c_n \otimes \rho(r_n) \quad (5)$$

A string  $x \in \Sigma^*$  (the asterisk denotes the Kleene closure) is accepted by  $T$  if there exists a successful path  $\pi'$  labeled with the input string  $x$  (i.e.,  $\iota(\pi') = x$ ). The weight associated by  $T$  to the sequence  $x$  is then the  $\oplus$ -sum of the weights of the successful paths  $\pi'$  labeled with the input label  $x$  and an output label  $y \in \Delta^*$ .

Such mapping from  $\Sigma^* \times \Delta^*$  to  $\mathbb{K}$  is called a weighted transduction of a given automaton  $T$  and is defined as

$$L_T(x, y) = \bigoplus_{\pi' \in q \xrightarrow{x, y} q'} \omega(\pi') \quad (6)$$

where  $\bigoplus$  represents the summation using the collection operator  $\bigoplus$  and  $\pi \in q \xrightarrow{x, y} q'$  denotes the set of paths from  $q$  to  $q'$  labeled with the input string  $x$  and the output string  $y$ .

The AT&T FSM Library offers software tools for operations with finite-state automata, such as, for example, union, concatenation and Kleene closure and also tools for automata determinization and minimization. Let us present the exact definitions of the two operations that are essential for the application of the finite-state transducers to speech recognition - projection and composition [5].

Each transduction  $L_T : \Sigma^* \times \Delta^* \rightarrow \mathbb{K}$  has two associated weighted languages - the first (input) projection  $\xi_1(L_T) : \Sigma^* \rightarrow \mathbb{K}$  and the second (output) projection  $\xi_2(L_T) : \Delta^* \rightarrow \mathbb{K}$  defined by

$$\xi_1(L_T)(x) = \bigoplus_{y \in \Delta^*} L_T(x, y) \quad (7)$$

$$\xi_2(L_T)(y) = \bigoplus_{x \in \Sigma^*} L_T(x, y) \quad (8)$$

A composition of two transductions  $L_T : \Sigma^* \times \Delta^* \rightarrow \mathbb{K}$  and  $L_S : \Delta^* \times \Gamma^* \rightarrow \mathbb{K}$  is defined by

$$L_R(x, z) = L_T(x, y) \circ L_S(y, z) = \bigoplus_{y \in \Delta^*} L_T(x, y) \otimes L_S(y, z) \quad (9)$$

where  $x \in \Sigma^*$ ,  $y \in \Delta^*$  and  $z \in \Gamma^*$ . The transducer  $R$  then represents a composition of the automata  $T \circ S$  and provides a mapping  $\Sigma^* \times \Gamma^* \rightarrow \mathbb{K}$ . It is clear that the composition is useful for combining different information sources or different levels of representation.

Using the composition, the speech recognizer can be represented by the so-called recognition cascade  $H \circ C \circ L \circ G$ , where each component is a weighted finite-state transducer over the tropical semiring -  $H$  represents an acoustic model,  $C$  transduces context-dependent phones to context-independent ones,  $L$  represents a pronunciation lexicon and finally  $G$  is a word-based language model. The decoder task of finding the best word sequence  $\hat{W}$  can be then expressed in terms of FST operations as

$$\begin{aligned} \hat{\pi}' &= \arg \min_{\pi'} \xi_2(O \circ H \circ C \circ L \circ G) \\ \hat{W} &= \iota(\hat{\pi}') \end{aligned} \quad (10)$$

where  $O$  is the an input sequence of acoustic features which can of course be transformed to a trivial finite-state machine as well. The  $C \circ L \circ G$  part of the cascade is constructed beforehand whereas the composition with  $O$  and  $H$  is performed during the decoder run.

### 3. Class-based models from the FST point of view

Let us consider a standard class based  $n$ -gram language model where a word can belong to more than one class. The probability of the word  $w_i$  given the history  $h_i$  is defined by

$$P(w_i | h_i) = \sum_{c_i} P(w_i | c_i) \cdot P(c_i | c_{i-n+1}, \dots, c_{i-1}) \quad (11)$$

where  $P(w_i | c_i)$  denotes the probability of the word  $w_i$  given the class  $c_i$  and  $P(c_i | c_{i-n+1}, \dots, c_{i-1})$  denotes the  $n$ -gram probability that the class  $c_i$  will follow the previous  $(n-1)$  classes  $c_{i-n+1}, \dots, c_{i-1}$ .

It is evident that the language model above represents a hidden Markov model just as other components of the recognition cascade. Thus it is appropriate to evaluate the probability  $P(w_i | h_i)$  within the same semiring and rewrite Formula (11) in a general form as

$$\hat{P}(w_i | h_i) = \bigoplus_{c_i} \hat{P}(w_i | c_i) \otimes \hat{P}(c_i | c_{i-n+1}, \dots, c_{i-1}) \quad (12)$$

which in the case of the tropical semiring and negative natural log probabilities results into

$$\begin{aligned} -\ln P(w_i | h_i) &= \min_{c_i} \{ -\ln P(w_i | c_i) + \\ &\quad + (-\ln P(c_i | c_{i-n+1}, \dots, c_{i-1})) \} \end{aligned} \quad (13)$$

Formula (11) is a special case of Formula (12) for the so-called probability semiring  $(\mathbb{R}, +, \cdot, 0, 1)$ . Many existing systems evaluate the probability  $P(w_i | h_i)$  within the log semiring which is the image by logarithm of the probability semiring. Formula (12) then becomes

$$\begin{aligned} -\ln P(w_i | h_i) &= \sum_{c_i} (-\ln P(w_i | c_i) + \\ &\quad + (-\ln P(c_i | c_{i-n+1}, \dots, c_{i-1}))) \end{aligned} \quad (14)$$

Note that the summation in (14) increases the computational load and causes inconsistency with the rest of the recognition cascade. Moreover, the performance gain of Formula (14) over Formula (13) has not been consistently proven. Therefore we prefer Formula (13) in our work.

Equation (12) corresponds to the transducer composition formula (9) and hence a class-based language model can be represented by a composition of two finite-state transducers  $T \circ V$  where  $T$  realizes a mapping from word-class pairs  $(w_i, c_i)$  to  $-\ln P(w_i|c_i)$  and  $V$  is a well-known  $n$ -gram transducer based on word classes instead of words.

We can simply replace the word-based language model  $G$  with  $T \circ V$  in the recognition cascade. In that case we obtain the best class sequence instead of the best word sequence in the output of the decoder. However, the AT&T decoder is built so that it produces not only the best sequence but also a so-called lattice. The lattice is an acyclic finite-state transducer containing the most probable paths through the recognition cascade for a given utterance. It has context-dependent phones on the input side and output labels from the rightmost transducer in the cascade on the output side.

Therefore we can retrieve the best word sequence  $\hat{W}$  even from the class-based lattice  $X$  using the following operations

$$\begin{aligned}\hat{\pi}' &= \arg \min_{\pi'} \xi_2(\xi_1(X) \circ C_{\bar{1}} \circ L_{\bar{1}}) \\ \hat{W} &= \iota(\hat{\pi}')\end{aligned}\quad (15)$$

where  $C_{\bar{1}}$  and  $L_{\bar{1}}$  are special variants of  $C$  and  $L$  with all weights set to  $\bar{1}$ . However, the transducer  $C \circ L \circ T \circ V$  often becomes too large due to the many-to-many word-to-class mapping. Thus the first recognition run is usually performed with a simple word-based  $n$ -gram  $G$ . Then the language model score is stripped from the output lattices and the resulting lattices are rescored with  $T \circ V$ . In terms of FST operations, the best word sequence  $\hat{W}$  is determined by

$$\begin{aligned}\hat{\pi}' &= \arg \min_{\pi'} \xi_1(\xi_2(X_{Acc}) \circ T \circ V) \\ \hat{W} &= \iota(\hat{\pi}')\end{aligned}\quad (16)$$

where  $X_{Acc}$  denotes the lattice with the acoustic score only.

It is generally known (and our preliminary experiments proved it) that class-based language models yield more robust probability estimates than word-based models but at the same time they have worse discrimination ability ("sense of detail"). Thus word-based and class-based language models are usually combined in some manner. The FST framework offers a natural way of model combination - we can simply retain the word language model score in the output lattices and then compose the lattices with  $T \circ V$ . We have found out empirically that better results are achieved when the  $P(w_i|c_i)$  component is omitted from a class-based model. Such model combination can be expressed formally as

$$\begin{aligned}\hat{\pi}' &= \arg \min_{\pi'} \xi_1(\xi_2(X) \circ T_{\bar{1}} \circ V) \\ \hat{W} &= \iota(\hat{\pi}')\end{aligned}\quad (17)$$

where again  $T_{\bar{1}}$  is the transducer  $T$  with all weights set to  $\bar{1}$ . Both word-based  $n$ -gram  $G$  and class-based  $n$ -gram  $V$  are often scaled with factors  $s_w$  and  $s_c$ , respectively. Scaling is performed by multiplying each transition weight in a transducer by a scaling factor. A proper choice of the scaling factors improves the recognition accuracy by several percent.

Finding the best word sequence according to (17) with scaled  $G$  and  $V$  corresponds to the usage of a language model

$$\begin{aligned}P(w_i|h_i) &= P^{s_w}(w_i|w_{i-l+1}, \dots, w_{i-1}) \cdot \\ &\quad \cdot P^{s_c}(c_i|c_{i-n+1}, \dots, c_{i-1})\end{aligned}\quad (18)$$

## 4. Word classes based on part of speech

In this chapter we describe an example of a word-to-class mapping - a class membership of a word is defined by its morphological tag [6]. Every tag 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 usage of the morphological tags is motivated by the following facts:

1. It is a good example of a many-to-many word-to-class mapping since many words have more than one possible tag. Note that such ambiguity does not have to be a result of a word having several possible different parts of speech - the difference can appear in any position of the tag.
2. Using such morphological information is suitable for language modeling of the Czech language since Czech makes an extensive use of agreement.

The strongest agreement is between a noun and its adjectival or pronominal attribute: they must agree in gender, number and case. There is also agreement between a subject (expressed by a noun, a pronoun or even an adjective) and its predicate verb in gender and number, and for pronouns, also in person. Verbal attributes must agree in number and gender with its related noun, as well as with its predicate verb (double agreement). Possessive pronouns exhibit the most complicated type of agreement - in addition to the abovementioned triple attributive agreement with the possessed thing they must also agree in gender and number with the possessor.

All those morphological categories (gender, number, etc.) are included in the morphological tags. Therefore there should exist some dependencies between adjacent tags that can be captured by an  $n$ -gram language model. The situation is complicated by the already mentioned relatively free word order. However, this liberty of word ordering is in many cases rather theoretical and there exists the common order subject-verb-object.

## 5. Experimental evaluation

### 5.1. Speech and text corpora

The Czech TV & Radio Broadcast News speech corpus was used for acoustic model training. This speech corpus consists of news broadcasted on 3 TV channels and 4 radio stations during the period February 1, 2000 through April 22, 2000. The corpus contains over 50 hours of audio data stored on 347 files, which yield about 26 hours of pure transcribed speech. The broadcast news corpus does not contain weather forecasts, sports news and traffic announcements. The signal is single-channel, sampled at 22.05 kHz with 16-bit resolution. More details and corpus statistics are given in [8].

22 hours of the transcribed material were used for the acoustic model training and 4 hours were put aside. 20 minutes of this put-off data were used for finding the optimum scaling factors (development data) and the rest for evaluating the system per-

formance (evaluation data).

For the language modeling purposes we used texts from the newspapers Lidové Noviny spanning the period 1991 through 1995. The corpus contains approximately 33 million tokens (650k distinct words). The data were processed by the Czech morphological analyzer [6] and the tagger [7] in order to obtain training data for tag-based class language models. There were only 1948 distinct tags in the corpus.

## 5.2. Acoustic models

The acoustic models were trained using HTK, the hidden Markov model toolkit [9]. The recognition system is based on a continuous density HMMs trained on approximately 22 hours of speech selected from the Czech TV & Radio Broadcast News corpus.

The speech features parameterization employed in training and test are the mel-frequency cepstra, including both the delta and the delta-delta sub-features; cepstral mean subtraction is applied to all features on a per utterance basis. Triphone state clustering was carried out using broad acoustic phonetic classes.

## 5.3. Experiments

We performed several experiments with various language models in order to evaluate the concepts presented above. All experiments were carried out using the AT&T decoder and/or AT&T FSM Library tools. Language models were estimated utilizing the SRILM toolkit [10]. In all cases the scaling factors were first experimentally optimized on the development data and then the best values were used for the recognition of the evaluation data. Numbers in Table 1 of course report the evaluation data accuracy.

The result of Experiment #1 is our baseline. We took 60k most frequent words from the Lidové noviny corpus and built a standard word bigram language model with the Katz's discounting. The out-of-vocabulary rate on the test set (the development-data and the evaluation-data together) was 8.27%. Then we ran the decoder, generated word lattices and evaluated the baseline accuracy. The rest of the experiments were done using the lattice rescoring technique.

In Experiment #2 we removed the original language model score from the lattices and rescored them with a word trigram model. Only trigrams that occurred 2 or more times were included in the model in order to keep a reasonable model size. Again the Katz's discounting was employed.

Experiments #3 and #4 involve the rescoring of the lattices with the standard tag-based bigram and trigram model, respectively (see Equation (12)). Again the baseline language model score was stripped from the lattices and the rescoring was performed according to Formula (16). The probability  $P(w_i|c_i)$  was estimated using the maximum likelihood approach whereas the  $n$ -gram probability  $P(c_i|c_{i-n+1}, \dots, c_{i-1})$  was computed within the Katz's backing-off scheme.

Finally, Experiments #5 and #6 represent the combination of the word-based and class-based language models as described by Formula (17).

## 6. Conclusions

It was shown in our paper that the widely used class-based language model with many-to-many word-to-class mapping can be consistently represented within the FST framework. We also proposed a technique for combining word-based and class-based language models.

Experiment - language model	Accuracy [%]
#1 - word bigram (baseline)	70.20
#2 - word trigram	71.12
#3 - standard tag bigram	69.12
#4 - standard tag trigram	69.86
#5 - word bigram & tag bigram	72.39
#6 - word bigram & tag trigram	72.73

Table 1: *Experimental results.*

Experiments with the class model based on the morphological tags showed that such a model combined with the standard word  $n$ -gram can improve the recognition accuracy by about 2% absolute.

The comparison of various bigram and trigram language models revealed that trigram models do not bring too much improvement over bigram models. This could be surprising; however, experiments with other Czech speech corpora yielded similar results and therefore this fact probably stems from the data sparseness that was already mentioned in the Introduction.

## 7. Acknowledgements

This work was supported by the project No. LN00A063 of the Ministry of Education in Czech Republic.

## 8. References

- [1] M. Mohri, F. Pereira and M. Riley, "Weighted Finite-State Transducers in Speech Recognition", Proceedings of ASR2000, International Workshop on Automatic Speech Recognition: Challenges for the Next Millennium, Paris, 2000.
- [2] E. Whittaker, "Statistical Language Modelling for Automatic Speech Recognition of Russian and English", Ph.D. thesis, University of Cambridge, 2000.
- [3] G. Riccardi, R. Pieraccini and E. Bocchieri, "Stochastic automata for language modeling", Computer Speech and Language 10 (1996), pp. 265-293.
- [4] W. Kuich and A. Salomaa, "Semirings, Automata, Languages", Springer-Verlag, Berlin, 1986.
- [5] F. Pereira and M. Riley, "Speech Recognition by Composition of Weighted Finite Automata", Finite-State Language Processing (Eds. E. Roche and Y. Schabes), MIT Press, 1997.
- [6] J. Hajič, "Disambiguation of Rich Inflection - Computational Morphology of Czech", Charles University Press - Karolinum. In press.
- [7] J. Hajič, P. Krbeč, P. Květoň, K. Oliva and V. Petkevič, "Serial Combination of Rules and Statistics: A Case Study in Czech Tagging", Proceedings of ACL 2001, Toulouse, 2001.
- [8] J. Psutka, V. Radová, L. Müller, J. Matoušek, P. Ircing and D. Graff, "Large Broadcast News and Read Speech Corpora of Spoken Czech", Proceedings of Eurospeech 2001, Aalborg, 2001.
- [9] S. Young et al., "The HTK Book", Entropic Inc., Cambridge, 1999.
- [10] A. Stolcke, "SRILM - an Extensible Language Modeling Toolkit", Proceedings of ICSLP 2002, Denver, 2002.