MORPHOLOGICAL RANDOM FORESTS FOR LANGUAGE MODELING OF INFLECTIONAL LANGUAGES

Ilya Oparin[1 2 3], *Ondřej Glembek*[3], *Lukáš Burget*[3], *Jan Černocký*[3]

[1]Dept. of Computer Science and Engineering, University of West Bohemia, Plzen, Czech Republic
[2]Speech Technology Center Ltd., St.Petersburg, Russia
[3]Speech@FIT, Brno University of Technology, Brno, Czech Republic

ABSTRACT

In this paper, we are concerned with using decision trees (DT) and random forests (RF) in language modeling for Czech LVCSR. We show that the RF approach can be successfully implemented for language modeling of an inflectional language. Performance of word-based and morphological DTs and RFs was evaluated on lecture recognition task. We show that while DTs perform worse than conventional trigram language models (LM), RFs of both kind outperform the latter. WER (up to 3.4% relative) and perplexity (10%) reduction over the trigram model can be gained with morphological RFs. Further improvement is obtained after interpolation of DT and RF LMs with the trigram one (up to 15.6% perplexity and 4.8% WER relative reduction). In this paper we also investigate distribution of morphological feature types chosen for splitting data at different levels of DTs.

Index Terms— Speech recognition

1. INTRODUCTION

In this paper we study the application of the decision tree (DT) and random forest (RF) approaches to language modeling of Czech as an inflectional language. The DT mechanism for estimating probabilities of words following each other has long been known [1] as an alternative to N-gram approach. In the pioneering work on DTs as LMs [1], improvement in perplexity was shown when DTs were used for a restricted recognition task with grammatical classes elaborated by hand. For more general and fully automated tasks the results were mostly discouraging and no steady improvement over N-gram LMs was reported. This can be explained by the peculiarities of DTs themselves. The DTs suffer from training data fragmentation and absence of theoretically founded grow-stopping criteria [2]. However, with the recent advances in language modeling that extended the use of decision trees to that of random forests, this direction of research was brought back to the spotlight [2],[3].

Decision trees were introduced into language modeling to alleviate the problem of data sparsity: with the help of DTs it is possible to cluster together similar histories at the leaves of a tree. Each leaf forms an equivalence class of the histories that share the same distribution over predicted words. *Predictor* refers to words in the particular position in N-gram history we ask *yes/no* questions in the node to split data when we propagate them down the tree. If the predictor is the previous word, a question looks like "Is the previous word in the set S or $\overline{S}$?". The data (i.e. N-grams) corresponding to *yes* answers are propagated through the left branch going out of a node, the *no*-data go to the right branch. Actually, a conventional N-gram model can be regarded as a special case of the tree model in which the set S consists of one individual word at each node. Ideally, at the training phase all possible predictors and questions should be tried at each node and the "best" predictor/question pair should be picked and stored for that node. However, in real life different solution is used, as we describe in the next section.

A random forest is a collection of DTs that include randomization in the tree-growing algorithm. The underlying assumption is that while one DT does not generalize well to unseen data, a set of randomized DTs might perform better. Greedy algorithms are used at the stage of DT construction for choosing best questions to split data. As a result, trees are only locally but not globally optimal (with respect to training data). Randomized trees are not locally optimal, but the collection of them may be closer to a global optimum and thus provide better results. When we randomize DTs to form a RF there are two basic sources of randomization: initial split of data in a node in two sets S, $\overline{S}$ and random selection of one of the predictors.

2. DECISION TREES AND RANDOM FORESTS

How to measure goodness of a question that splits data in the node? A DT is constructed in a way to reduce the uncer-

This work was partly supported by Ministry of Trade and Commerce of Czech Republic under project FT-TA3/006 and by Ministry of Interior of Czech Republic under project VD20072010B16. The hardware used in this work was partially provided by CESNET under project No. 201/2006. Lukáš Burget was supported by Grant Agency of Czech Republic under project No. GP102/06/383.

tainty about the event predicted. Thus, entropy can be used as the goodness measure. One should measure entropy for data M in a node before split, then split data in two sets S and $\overline{S}$ according to *yes/no* questions and find the entropy reduction under the split. The reduction ξ in entropy on development data shows how good the question is.

How to find the best question? It is unfeasible to try all possible questions that give rise to different data splits even for a moderate number of parameters. Regarding sizes of vocabularies used in real recognition tasks, that means that less extensive sub-optimal greedy approaches should be considered. There are a number of different variations of greedy tree-growing algorithms [1],[4]. They all suffer from the same drawback we already noted, that is local optimum is searched for splitting the histories. This problem is partially solved by the RF approach.

How to define a stopping criterion so that the node should be turned to a leaf? Entropy of a tree can always be decreased by increasing the number of leaves. However, such a tree will not be able to generalize on unseen data and will be characterized by high entropy on test data [4]. Minimum entropy reduction or minimum data thresholds may be used to turn a node into a leaf. However, these constants are empirical as opposed to measuring the entropy reduction under the same split on heldout data. If heldout data entropy is not reduced that means the split will lead to a tree that is strongly biased to the training data.

In this study, we generally accept the approach proposed in [4]. We use the *exchange algorithm* to find best questions in DT nodes and heldout data entropy reduction as stopping criterion. For RFs the exchange algorithm is initialized with data randomly split in sets S and $\overline{S}$ and a randomly chosen predictor to ask questions about.

3. MORPHOLOGICAL DECISION TREES

3.1. Morphological Features as Predictors

In word-based DTs questions like "Does the previous word belong to the set of words $\{ w_b, w_f, w_q, \ldots \}$?" are asked at each node. In morphological DTs we want to ask questions about morphological features of word predictors. We expect it to be particularly useful for morphologically-rich languages as Czech. In this study morphological feature types are word-form itself (W); word lemma, i.e. initial form of the word (L); word stem (S); part-of-speech - POS (P); full morphological tag (T) and inflection (I). Thus, the questions may be in the form "Is the full morphological tag of the predictor *animate singular noun in accusative case*?". Each combination of position in word history and morphological feature type is considered as an individual morphological predictor. Thus, in the process of tree construction, the tree-growing algorithm is given total freedom to choose independently between morphological features. An HMM-based Czech POS tagger developed at IFAL in Prague (http://ufal.mff.cuni.cz/) is used for tagging.

3.2. Smoothing

When traversing the DT, each node splits the data into two subsets, causing data sparsity. Navratil [5] proposes a smoothing technique, where the smoothed probability of a symbol s in a node is given as:

$$\hat{P}'_{sm}(s|l) = b_{s,l}\hat{P}(s|l) + (1 - b_{s,l})\hat{P}_{sm}(s|parent(l)) \quad (1)$$

where $b_{s,l}$ is defined in equation 2. The equation is applied recursively until $parent(l) = root$, when $\hat{P}_{sm}(s|root) = \hat{P}(s|root)$

$$b_{s,l} = \frac{C(s|l)}{C(s|l) + r} \quad (2)$$

where $C(s|l)$ stands for counts of words s in leaf l, and r is an empirical smoothing factor controlling the strength of the update. This kind of smoothing does not take account of discounting and backoff techniques that proved very helpful in N-gram based language modeling (e.g. Kneser-Ney). However, it is of interest to try this way of smoothing that showed good results for the automatic language identification task [6] for the purpose of language modeling in LVCSR.

3.3. Morphological Random Forests

There are two basic sources of randomization: initial data split in two sets S, $\overline{S}$ and random predictor selection. Initial split randomization is basically the same for word and morphological DTs. As for predictor position selection, randomization possibilities become much reacher for morphological DTs. Thus, if we work on trigram level and use 6 morphological features attached to each word, we end up with $2 * 6 = 12$ morphological predictors. At the same, time morphological features of different types (stem, inflection, POS, etc) are not equally important to predict next words. Thus if we randomly pick up only one of those, we have equal chances to pick up predictors pertaining to types with low predicting power and end up with very shallow trees. So we form a pool of "good" predictors at each node and then choose randomly one of them. In our study we form a pool of predictors that are above threshold p that is calculated according to a very simple formula

$$p = \frac{(e_{best\_red} - e_{worst\_red})}{100\%} \times n\% + e_{worst\_red} \quad (3)$$

where e_{best_red} and e_{worst_red} are the largest and smallest entropy reductions gained with some of the predictors. Value of n is an empirical constant, in our study it equals to 70.

Table 1. *Perplexity for stand-alone models.*

Model	IRP	ISS	MUL
Trigram	317	212	258
Word DT	433	253	336
Morph DT	413	252	320
Word RF	360	221	280
Morph RF	298 (6.0%)	190 (10.4%)	237 (7.4%)

Table 2. *Perplexity for interpolated models.*

Model	IRP	ISS	MUL
Trigram	317	212	258
Word DT	302 (4.7%)	198 (6.6%)	245 (5.0%)
Morph DT	296 (6.6%)	197 (7.1%)	240 (7.0%)
Word RF	292 (7.9%)	191 (9.9%)	234 (9.3%)
Morph RF	272 (14.2%)	179 (15.6%)	220 (14.7%)

4. EXPERIMENTS

4.1. Experimental Setup

The recognition of spoken lectures held in Czech is our target task. The transcriptions of three lectures on different subjects in the domain of information technology were chosen as the testing data: IRP (16K words), ISS (6K words) and MUL (10K words).

Small corpus *annot* of manually annotated lecture transcripts consists of 200K words. It is very close in topics and style to the testing data. Written corpus *lect* of lecture materials consists of 7.1M words, 240K words constituting the vocabulary. Both these corpora are used to build bigram LMs used to generate 500-best lists that are subsequently rescored with more sophisticated (DT and RF) models. For the best baseline LMs corresponding to both corpora were interpolated with the weights tuned according to perplexity. Joint 240K vocabulary is used for recognition.

Corpus *annot* is used as training data for DTs. The vocabulary was chosen as an intersection of *annot* and *lect* vocabularies and it consists of 15K words. Word history length is two, we thus work on the word trigram level. Open vocabulary models are built. At the stage of N-best list rescoring, probabilities assigned to unknown words are discounted with an empirical constant. Heldout data are used as stopping criterion. After the tree is grown, heldout data is poured down the tree to the leaves so that the resulting statistics correspond to the whole corpus. Each RF consists of 100 randomized DTs. In addition, minimal data threshold is also used to stop growing a tree, that is found empirically.

4.2. RESULTS

4.2.1. Perplexity

First we evaluate the performance of different LMs with perplexity. Since our training data is very close in topics and style to the testing data, the results give insight in real performance of the models even though the size of the training data is small. Table 1 represents perplexities for stand-alone models on three different testing lectures. If the improvement over the trigram model is gained, it is shown in brackets. We can see that individual DTs perform worse than the standard Kneser-Ney trigram model (trained with SRILM toolkit). Word RF does not show steady perplexity improvement on its own but rather performs in the same way as the trigram model. Little improvement of 2.6% for ISS data can not be considered noteworthy for perplexity experiments. This difference from results reported in [4] can be explained by the fact that in our framework we do not make use of any smoothing and backoff technique that are known successful for language modeling. However, with morphological trees we achieve a noteworthy improvement of perplexity over 10%.

Perplexity results after the interpolation of the trigram model with different DT-based ones are presented in Table 2. All DT-based models show steady perplexity improvement in interpolation with the trigram model.

Table 3. *Word accuracy and relative WER reduction for stand-alone models.*

Model	IRP	ISS	MUL
1-best	63.1	70.2	58.3
Trigram	63.8	70.9	59.2
Word DT	63.8	70.7	59.1
Morph DT	64.1	69.7	59.1
Word RF	64.2 (1.1%)	70.9	59.2
Morph RF	64.7 (2.5%)	71.9 (3.4%)	59.7 (1.2%)

4.2.2. Word Accuracy Estimation

Word accuracy for different stand-alone models is shown in Table 3. Row *1-best* corresponds to the 1-best accuracy for 500-best lists without any rescoring. Trigram LM is taken as the baseline. Relative WER improvement over the trigram model is shown in brackets, if gained. Following the results represented in Table 1, individual DTs do not directly improve the accuracy. However, both morphological and word RFs do. Table 4 shows results for the DT-based models after interpolation with the trigram one. The difference with the perplexity results presented in Table 2 is mostly in the lower improvement of the results with the interpolation of RFs.

4.2.3. Distribution of Morphological Predictors

We wanted to study the regularities how morphological predictors of different types are chosen at different tree-levels by the tree-growing algorithm. Distribution of predictors that

Table 4. *Word accuracy and relative WER reduction for interpolated models.*

Model	IRP	ISS	MUL
Word DT	64.5 (2.0%)	71.5 (2.1%)	59.6 (1.0%)
Morph DT	64.5 (2.0%)	71.3 (1.4%)	59.8 (1.5%)
Word RF	64.5 (2.0%)	71.6 (2.4%)	59.8 (1.5%)
Morph RF	64.8 (2.8%)	72.3 (4.8%)	60.1 (2.2%)

Table 5. *Distribution of morphological predictors related to -1 (previous) word at different tree depth in a RF (in per cents).*

Depth	W	S	L	T	P	I	Total
1	29.0	23.0	20.0	26.0	2.0	-	100.0
2	35.5	23.0	24.0	12.5	5.0	-	100.0
3	27.2	26.2	21.8	13.2	8.5	-	97.0
4	26.6	24.5	21.2	12.8	6.0	-	91.1
5	24.8	22.0	17.9	13.4	5.1	0.9	84.2
6	18.7	17.6	14.8	9.4	3.6	1.5	65.5
7	15.6	14.2	12.5	6.7	2.7	1.2	52.9
8	12.4	11.6	10.8	5.2	1.5	1.4	43.0
9	11.5	9.7	8.7	4.6	1.7	1.7	38.0
10	9.4	8.6	7.9	4.8	2.0	2.2	34.9

refer to -1 (i.e. previous) word in history at different node depths up to 10th in the morphological RF is presented in Table 5. Distribution that refer to -2 position is skipped due to the lack of space in this paper. Rows correspond to to the depth of nodes in the tree. Columns correspond to morphological predictor types. Summary statistics for -1 predictors is represented in the last column.

The distribution of morphological features chosen at each node in the tree at the training phase shows some regularities. First, in upper nodes questions about morphological features of the previous word are chosen. Then, as moving down the tree, -2 word comes to the foreground. Second, feature types that cover narrower classes (W, S, L, T) are chosen at the upper levels in the tree. Questions about features that cover wider classes (POS, I) tend to appear more and more often as we go deeper down the tree.

5. DISCUSSION AND FUTURE WORK

Random forests consisting of randomized DTs that take account of grammatical information outperform word-based RFs for the Czech language. However, this does not hold for individual DTs. This fact may lead to the conclusion that morphological RF are likely to perform better not due to the morphological information itself but rather to wider randomization possibilities.

There are several directions of the future work. First, DTs and RFs on larger data should be trained and tested. Second, we plan to run similar experiments for other inflectional languages. Third, while extension of context to more than two previous words did not give much positive result for English [4], the situation may be different for inflectional languages with relatively free word order. Finally, integration of morphological predictors with smoothing and backoff techniques developed for N-grams should further improve the results.

6. CONCLUSIONS

In this paper we studied language modeling of an inflectional language with decision trees and random forests for recognition of spoken lectures. Both approaches were tested with taking different sources of information into account: pure lexical and morphological. Our experiments proved that decision trees do not outperform classical trigram model. The perplexity and WER improvement is possible only with the interpolation of tree-based models with an trigram one. Random forests, on contrary, directly improve the baseline. Even larger improvement is gained with the interpolation of RFs with the conventional trigram model.

The distribution of different types of morphological predictors in a RF show that questions about predictors corresponding to the previous word are preferred at top levels of DTs, while those corresponding to the more distant word position gradually get prominence as we traverse DTs in top-down manner. Feature types that cover narrower classes (word, stem, lemma, full morphological tag) are mostly chosen at the upper levels in the tree.

7. REFERENCES

[1] L.R. Bahl, P.F. Brown, P.V. de Souza, and R.L. Mercer, "A tree-based statistical language model for natural language speech recognition," *Computer, Speech and Language*, vol. 37, pp. 1001–1008, July 1989.

[2] P. Xu and F. Jelinek, "Random forests in language modeling," in *Proceedings of EMNLP'04*, 2004, pp. 325–332.

[3] Su Y., Jelinek F., and Khudanpur S., "Large-scale random forest language models for speech recognition," in *Proceedings of Interspeech-07*, 2007.

[4] P. Xu, *Random Forests and the Data Sparseness Problem in Language Modeling*, Ph.D. thesis, John Hopkins University, 2005.

[5] J. Navrátil, Q. Jin, W. Andrews, and J.P. Campbell, "Phonetic speaker recognition using maximum-likelihood binary-decision tree models," in *Proceedings of ICAASP'03*, 2003.

[6] O. Glembek et al., "Advances in phonotactic language recognition," in *Submitted to Interspeech'08*, 2008.