Over-Generative Finite State Transducer n-gram for Out-Of-Vocabulary Word Recognition

Ronaldo Messina and Christopher Kermorvant
A2iA S.A.
39 rue de la Bienfaisance
Paris 75008 France
rm,ck@a2ia.com

Abstract—**Hybrid statistical grammars both at word and character levels can be used to perform open-vocabulary recognition. This is usually done by allowing the special symbol for unknown-word <unk> in the word-level grammar and dynamically replacing it by a (long) n-gram at character-level, as the full transducer does not fit in the memory of most current computers.**

We present a modification of a finite-state-transducer (fst) n-gram that enables the creation of a static transducer, i.e. when it is not possible to perform on-demand composition. By combining paths in the "LG" transducer (composition of lexicon and n-gram) making it over-generative with respect to the n-grams observed in the corpus, it is possible to reduce the number of actual occurrences of the character-level grammar; the resulting transducer fits the memory of practical machines.

We evaluate this model for handwriting recognition using the RIMES and the IAM dabases. We study its effect on the vocabulary size and show that this model is competitive with state-of-the-art solutions.

I. Introduction

The number of words that are possible to be encountered by a handwriting (or speech) recognition system is quite large and collecting data to reliably estimate the parameters of statistical grammars for all the possible words is not really feasible. One alternative to circumvent this problem is to use a hybrid grammar, where a word-level n-gram is used for the most common words and a character-level n-gram (with a high value for n) is employed to model the other words [1]. By limiting the vocabulary, it is possible to insert a special symbol in the word-level grammar for the unknown words (typically <unk>), which is replaced by the character-level n-gram during decoding.

When finite-state-transducers (fst) [2] are used in the recognition system, it is not possible to statically compose (complete the composition in the machine's memory) the lexicon and the grammar, as the resulting transducer would be huge (above 1 Terabyte). This is caused by the relatively large number of occurrences of <unk> in the grammar, depending on how many words are in the vocabulary; each occurrence of <unk> results in an instantiation of the whole character-level n-gram (which has to be quite large, to capture the most salient character sequences in the language).

In this paper, we propose to combine some paths in the transducer representing the composition of the lexicon and the word-level grammar, leaving a place-holder which is replaced by the character-level n-gram (composed with a character-lexicon). The idea is to combine all paths that pass through <unk> , so fewer actual instances of the character-level grammar are needed; this results in a over-generative grammar, as paths not existing in the original grammar are created. Note that this model theoretically invalidates some of the back-off values as we create paths not present in the language model, but we experimentally show that it has no or little effect on recognition performance.

The remainder of this paper is organized as follows: at first we describe the recognition system used, then the over-generative approach to construct the fst is presented. This is followed by a description of the databases used to assess the approach, followed by the experiments performed. We experiment on the combination of paths and also on the effect of the vocabulary size. Finally we conclude with some remarks on future work.

II. Recognition system

Our recognition system uses a Recurrent Neural Network (RNN) [3] for the optical model and the Kaldi [4] decoder, based on finite-state-transducers. The RNN used (in-house implementation) is 2D and multidirectional, scaning the image in four different directions [5], producing predictions for each character in the alphabet. These predictions are decoded by Kaldi using a composed fst ("HCLG" as in the usual recipe [2].) This fst imposes lexical and syntactical constraints on the RNN's predictions and a decoding procedure finds the most likely string of words. RNN parameters are trained using Stochastic Gradient Descent [**?**]: a model update happens after each training sample (each line of characters) is visited. The learning rate is constant, and was fixed to 0.001 in all epochs.

There are a few hyper-parameters that can be tuned: a beam size that controls the pruning of hypotheses, a weight on the prior probabilities of the characters, and a weight on the optical predictions to balance its dynamic range with the one of the n-gram fst. In the experiments performed there was no tentative to tune those parameters; the beam size helps controlling the decoding-time and a tight value was arbitrarily determined (we run some experiments on a restrited test set and stop increasing the beam when the performance did not improve any more). The other weights were also arbitrarily set to some practical values, from previous experiments.

Next we describe the method used to construct over-generative transducers for OOV modeling.

III. Over-generative fst for OOV modeling

We follow the approach described in [1] to create hybrid language models (word-level and character-level). The vocabulary is limited to the top K most frequent words and a character-level n-gram (we used $n = 10$) is trained on the words present in the corpus but not in the given vocabulary; their frequency is preserved in the estimation of the n-gram (i.e. if a OOV word is present ten times in the corpus, it is counted ten times in the character n-gram). Note that it is also possible to include more data from other sources to enrich the character-level language model, enforcing the morphological constraints of the language.

The words in the corpus not present in the limited vocabulary are replaced by a special symbol (unknown-word) <unk> in the language model; during decoding its place is taken by the character-level n-gram. The frequent words in the language are represented by their instances in the word-level language model, while the open part of the vocabulary is modelled by the character-level n-gram.

To make it possible to statically construct the decoding transducer, we introduce a path merging procedure that limits the number of instances of the character-level n-gram in the hybrid language model.

The following options are experimentally evaluated in this work:

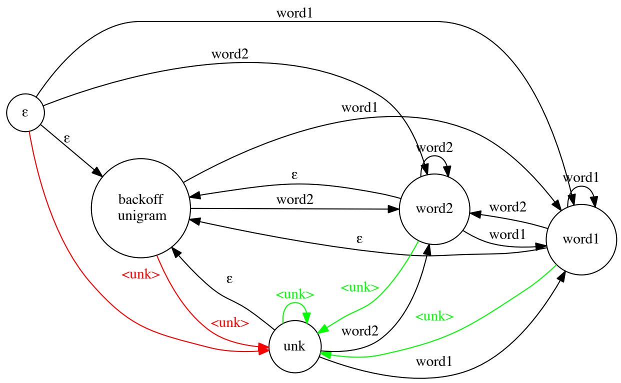
- Merge all paths through the unknown word into a single instance;
- Merge the paths if the unknown word appears as a unigram, then all paths if present in a bigram and finally the same for trigrams; resulting in three instances; and
- The same as the previous, but for bigram and trigrams, keep the instance dependent on the position, so there are two instances for bigram (left and right) and three for trigrams (left, middle, and right); this amounts to six instances.

The last point is illustrated, for a bigram only (for space reasons), in figures 1a before applying the procedure and in figure 1b after its application. The red paths are for <unk> at the left-hand side of the bigram, while the green ones when it appears at the right-hand side.

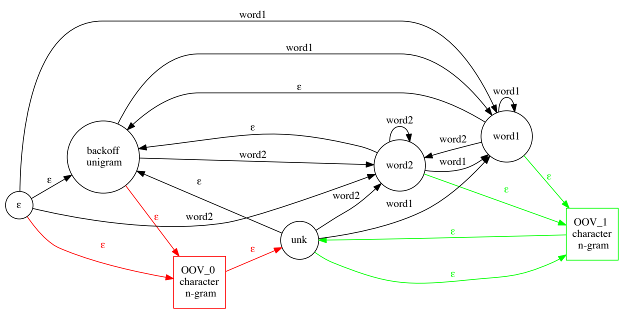
This procedure creates paths that do not exist in the original word-level n-gram ($n = 3$): from any given entry point in the unknown-word, all other paths through the other n-gram containing <unk> become reachable. This is the same as entering into a given n-gram during decoding but going out via a different n-gram, due to the path combination through the unknown-word. One could argue that the resulting model does not have the probabilities that sum up to one (but they could be re-scaled), but there is no requirement for this by the decoding process; moreover, the probabilities estimated for unknown-words are not expected to be reliable. After visiting an unknown-word, it makes little sense to restrict to the seen contexts directly reachable, as this word could be anything. One way of interpreting the merging is that the word history is reset when passing through the unknown-word.

IV. Databases

Two databases are used in the experiments, one in English [6] (IAM) and the other in French [7] (Rimes). The data in IAM are handwritten sentences from the LOB corpus [8], while Rimes contains handwritten letters. One optical model is trained for each database using stochastic gradient descent on the RNN.



(a)



(b)

Fig. 1. Bigram before (a) and after (b) the merging procedure.

To be able to compare the results on the IAM database with those in [1], we used the same data to train the language model; the following corpora were used: LOB [8], Wellington [9], and Brown [10]. The data used to train the language-model amounts to 3273290 words for a vocabulary of about 106k words. Training data for the optical model contains 81681 words, with a vocabulary of 10321 words.

The Rimes database comprises 90618 words (with a vocabulary of 6642 words) for training both the optical and the language model.

In the IAM test corpus, there are 336 pages, amounting to 25753 words, from which 5373 are unique; for the Rimes database (72 pages) the test set has 4920 words (with 1221 unique words). We used the same set as for the word recognition task of Rimes, but on the complete lines (the language model is not on paragraphs); the usual test set for IAM was used.

V. Experiments and Results

To asses the performance of the over-generative transducers, firstly we examine the effect of the number of the character-level n-gram instances, then we limit the vocabulary at different sizes to verify how the OOV model deals with a larger number of words not present in the word-level language model. In table I we present some results on these databases presented elsewhere [11], [**?**], [1]; there is a gap in the performance in some results due to a different optical model. The number between parentheses is the size of vocabulary.

TABLE I. Baseline results on IAM and Rimes.

Database	Reference	Vocabulary	%WER
IAM	[1]	closed (20k)	22.2
		open	17.3
Rimes	[**?**]	closed (6k)	15.2
	[11]	closed (6k)	29.4
		open	27.2

Using the IAM database and a vocabulary comprising the top 20k most frequent words, the word and character error rates in table II are obtained; we include the results for a grammar without the OOV part to assess the gain provided by the open-vocabulary approach and to illustrate the relative sizes of the compositon of lexicon and grammar, we indicate the size of "LG" (as in the usual recipe [2]).

TABLE II. %WER and %CER for the different number of instances of the character-level n-gram on the IAM database.

#char-level n-gram instances	%WER	%CER	LG (MB)
0	24.5	10.3	96
1	23.5	9.4	120
3	22.4	9.1	158
6	19.6	8.5	212

As expected, performance increases when the path merging becomes less aggressive; the gain with respect to the situation where there is no OOV modeling is at the same order (about 5% absolute WER reduction) as reported in [1] using the on-demand composition (the lexicon and grammar are composed as needed during decoding) and a vocabulary with the top 20k most frequent words. There is a larger gap when going from three to six instances of the unknown-word (modeled by the character-level n-gram). This can be explained because in the latter, the merged paths are not really crossing the boundaries of the n-grams, which is not the case when there are less than six separate instances.

Regarding the training data for the word-level language model, table III shows the percentage of the words in the corpus covered by each dictionary size. Half of the mass (number of occurrence of words) of the training data is represented by 92 words in the IAM corpus and by 43 on Rimes.

TABLE III. TRAINING CORPUS COVERAGE FOR THE DIFFERENT VOCABULARY SIZES.

IAM		Rimes	
Vocabulary	%Corpus	Vocabulary	%Corpus
5k	85.6	1k	88.9
10k	90.9	1.5k	91.9
15k	93.4	2k	93.7
20k	94.8	2.5k	94.9
25k	95.9	3k	96.0

To verify the effect of the vocabulary size on the performance, we limited the vocabulary from a quite restrictive 5k words to 25k words for the IAM database, and from 1k to 3k words for the Rimes corpus. The results are given in tables IV and V, for grammars with and without OOV modeling (baseline in the table); the OOV rate is also given for each vocabulary size.

TABLE IV. %OOV AND WORD ERROR RATE FOR DIFFERENT VOCABULARY SIZES ON IAM.

Vocabulary	%OOV	%WER (baseline)	%WER
5k	14.7	36.6	24.6
10k	10.2	29.9	21.0
15k	8.0	26.7	20.0
20k	6.7	24.5	19.6
25k	5.7	23.4	19.4
full (106k)	2.1	19.1	n/a

There is not a large gap in performance going from 10k to 20k vocabulary (1.4% absolute for double vocabulary size) in the IAM corpus and the performance with the full vocabulary is slightly better, but this is due to fewer errors in the in-vocabulary part of the test corpus.

TABLE V. %OOV AND WORD ERROR RATE FOR DIFFERENT VOCABULARY SIZES ON RIMES.

Vocabulary	%OOV	%WER (baseline)	%WER
1k	12.8	24.5	14.4
1.5k	10.0	20.9	13.8
2k	8.3	18.6	13.7
2.5k	7.4	17.4	13.5
3k	6.9	16.8	13.3
full (6.7k)	5.1	14.6	n/a

In the Rimes database, the performance stabilizes for vocabularies above 1.5k words (this is mainly due to the restricted size of the corpus used to train the language model). The performance is better than the one with the complete vocabulary.

The gain in performance when using the OOV modeling is higher for smaller vocabularies, but there is a limit on the capacity to recover the OOV words using the hybrid grammar. Ultimately, there is a balancing act between the capacity of the word-level language model to hypothesize word sequences containing unknown words and the ability of the character-level n-gram to correctly recognize the sequence of characters composing the word.

As in most corpora, a relatively restricted number of words accounts for the major part of the text. If a guideline on the number of words to limit the vocabulary is needed, take as many words as to cover 95% of the corpus, which seems to be a good compromise. This leaves an entry point for unknown words without taking too much probability mass from the observed data. The capacity of the hybrid grammar to deal with unknown words is a compromise between the morphology of the OOV that are encountered in practice and the one represented in the character n-gram.

If we analyze the OOV words in both test corpora, in IAM there are many proper nouns, a few dates and quite a few are regular words with the first letter capitalized. In the Rimes corpus, there are many code-like words like license plates or numbers that are OOV, and it seems that the amount of proper names is much lower; we also notice some words that are all capitalized. This database consists of letters, but the headings (names and addresses) are not part of the transcribed data that was made public; we can expect a higher OOV ratio when these fields are included in the openly distributed dataset.

Regarding the size of the transducer, there is a limitation due to the size of the character-level n-gram. As it is copied at most six times, the resulting graph can be quite large, so some pruning of the n-gram is needed. But its size is still a tiny fraction of the full transducer if static composition was used.

As new paths are added in the grammar fst, it is possible that some paths through the back-off state are redundant, but we did not experiment with the pruning of those paths to reduce the size of the transducer.

VI. CONCLUSION

This paper presents a over-generative grammar for unknown words, based on hybrid word/character level n-grams, enabling static composition when finite-state transducers are used in the decoder. The word-level n-gram is made over-generative as paths going through the unknown-word are merged so that there are as few instances of the character-level n-gram as possible (six gave the best results).

Experimental results showed that the approach gives practical results that are in line with what can be expected from the hybrid modeling, with on-demand composition. There is no noticeable degradation due to the over-generative nature of the resulting grammar. Our results are a bit worse than those presented in [1]; as the pre-processing and optical model are different, we cannot directly compare the results to say if the approximation of the static composition is the cause of the difference; but the gain with respect to no OOV modeling is quite similar.

The presence of the character-level n-gram effectively deals with most of the OOV words in the test data, under the conditions that the statistics can represent the morphology of those words. It is very difficult to achieve good performance in this conditions if the OOV are very specific such as foreign words, proper names, alphanumeric codes, etc. It is time consuming to run an analysis on the error regarding the type of OOV and we did not perform it.

Having an entry point for unknown words and the possibility to "plug-in" a character-level n-gram at this point, allows other kinds of grammar to be cast onto the fst formalism to deal with new words. The presented approach works when the OOV words are not very specific (specific OOV examples would be numbers or dates); in those specific cases, as the context is quite well-defined, using class-grammars should be superior.

An alternative way to deal with some OOV that are caused by capitalization is to allow capitals as alternative in the first letter, but this would imply in some adaptations in the language model to accommodate the different spellings for the words; this would be trivial in speech recognition where the issue of capitalization producing different words does not exist. Another issue that is present in written text recognition is the punctuation (which is either not present in language models for speech), that could generate an OOV if a comma or full stop is "glued" to the word. The latter is much harder to be dealt with the character-level n-gram.

Acknowledgement

This work was funded by the French Grand Emprunt-Investissements d'Avenir program through the PACTE project.

References

[1] M. Kozielski, D. Rybach, S. Hahn, R. Schlüter, and H. Ney, "Open vocabulary handwriting recognition using combined word-level and character-level language models," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 8257–8261, May 2013.

[2] M. Mohri, "Finite-state transducers in language and speech processing," *Computational Linguistics*, vol. 23, pp. 269–311, 1997.

[3] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *NIPS*, 2008, pp. 545–552.

[4] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[5] T. Bluche, J. Louradour, M. Knibbe, B. Moysset, F. Benzeghiba, and C. Kermorvant, "The a2ia arabic handwritten text recognition system at the openhart2013 evaluation," in *Submitted to DAS 2014*, 2014.

[6] U.-V. Marti and H. Bunke, "The IAM-database: an english sentence database for offline handwriting recognition." *IJDAR*, vol. 5, no. 1, pp. 39–46, 2002.

[7] E. Grosicki and H. El-Abed, "ICDAR 2011: French handwriting recognition competition," in *Proc. of the Int. Conf. on Document Analysis and Recognition*, 2011, pp. 1459–1463.

[8] S. J. Johansson, E. S. Atwell, R. Garside, and G. Leech, *The Tagged LOB Corpus: User's Manual*, The Norwegian Centre for the Humanities, Bergen, 1986.

[9] L. Bauer, *Manual of information to accompany the Wellington corpus of written New Zealand English*, Victoria University of Wellington, Wellington, 1993.

[10] W. N. Francis and H. Kucera, "Brown corpus manual," Department of Linguistics, Brown University, Providence, Rhode Island, US, Tech. Rep., 1979. [Online]. Available: http://icame.uib.no/brown/bcm.html

[11] C. Oprean, L. Likforman-Sulem, A. Popescu, and C. Mokbel, "Using the Web to create dynamic dictionaries in handwritten out-of-vocabulary word recognition," in *Proc. of the Int. Conf. on Document Analysis and Recognition*, 2013, pp. 989–993.