# Adding new words into a language model using parameters of known words with similar behavior

Luiza Orosanu, Denis Jouvet
Speech Group, LORIA
Inria, Villers-lès-Nancy, F-54600, France
Université de Lorraine, LORIA, UMR 7503, Villers-lès-Nancy, F-54600, France
CNRS, LORIA, UMR 7503, Villers-lès-Nancy, F-54600, France
Email: {luiza.orosanu, denis.jouvet}@loria.fr

***Abstract*—This article presents a study on how to automatically add new words into a language model without re-training it or adapting it (which requires a lot of new data). The proposed approach consists in finding a list of similar words for each new word to be added in the language model. Based on a small set of sentences containing the new words and on a set of n-gram counts containing the known words, we search for known words which have the most similar neighbor distribution (of the few preceding and few following neighbor words) to the new words. The similar words are determined through the computation of KL divergences on the distribution of neighbor words. The n-gram parameter values associated to the similar words are then used to define the n-gram parameter values of the new words. In the context of speech recognition, the performance assessment on a LVCSR task shows the benefit of the proposed approach.**

***Keywords*—speech-to-text transcriptions, language modeling, OOV words, similar words, part-of-speech tags, lemmas**

## I. Introduction

One of the main drawbacks of speech recognition systems is their incapacity to recognize words that do not belong to their vocabulary. Given the limited amount of speech training data, and also the limits in memory size and computational power that are imposed by any automatic speech recognizer, it would be impossible to conceive a system that covers all the words. Out-of-vocabulary (OOV) words will always be encountered, regardless the size of the vocabulary or the numerous general domains on which we train the language models (LM). As a result, when an OOV word is pronounced the speech recognition system will be forced to replace it with other short in-vocabulary words that are acoustically similar [1], [2], [3].

A large vocabulary continuous speech recognition (LVCSR) system can be very efficient when it's applied on similar domains to those on which it was trained. However, when there is a mismatch between the training data and the application domain, all the frequent words specific to the new domain will be treated as OOV words.

There are several options for solving this problem. The most classic solution suggests an adaptation of the language model [4], in order to learn the n-gram probabilities associated to the new words. But the adaptation can not be successful if there is not enough suitable data associated to the new words (such data is not always easy to find).

A different approach would use the data available on-line to estimate frequencies of unseen bi-grams [5].

Another solution would be to use class-based language models [6]. In this case, a new word has to be assigned to a predefined class. In the past, scientists would let the user decide to which class an unknown word belonged to [7]. But this can also be achieved automatically, based on the similarity between the new word and the in-class words, which can be estimated with the frequency of co-occurrence between two words [8], morphological tags [9] or the cosine-similarity between word vectors (which model the relation between words) [10]. Other class based language models that have enough general classes can even effectively model unseen events [11]. A class based unigram model can also be used to assess unigram probabilities for unseen words (based on morpho-syntactic similarity) which will then be included in a baseline language model [12].

A solution on word-based language models that uses the similarity between words has been studied in [13]. They search for specific n-gram sequences in which the OOV words are most semantically consistent and they keep the most frequent ones. Then they compute the conditional probabilities of solely those sequences, which are afterwards added to the language model.

Similarity measures based on word cooccurrence have been studied in [14] with a different objective: to create similarity-based language models. Their proposed methods make use of the Kullback-Leibler (KL) divergence, Jensesn-Shannon divergence, L1 norm and confusion probabilities.

Our alternative solution proposes to directly define and add to a word-based language model the n-gram entries of new words, based on the similarity of the new words with in-vocabulary words. Our approach is based on similar neighbor distributions and it requires very little data related to the new words (5 sentences for each new word are sufficient to achieve good results). We start by searching for words similar to the new words. Two data sets are thus required: a small list of sentences containing the new words (to establish a minimal list of usual neighbors) and a set of n-gram counts where to search for their similar words (i.e. words having similar neighbor distributions). For more consistent results, both data sets are tagged with their corresponding "word|part-of-speech" and "lemma|part-of-speech" units. The (dis)similarity measure between two words is defined as the KL divergence of their neighbor distributions. Once the list of similar words is de-

fined, all their n-gram probabilities (unigrams, bigrams and trigrams) are transposed on the new words.

The paper is organized as follows: Section 2 is devoted to the description of the data and tools used in our experiments, Section 3 provides a description of the methodology used to find similar words and to infer the n-gram parameters associated to new words, and Section 4 presents and analyzes the results.

## II. Experimental setup

### A. Data

The speech corpora used in our experiments come from the ESTER2 [15] and the ETAPE [16] evaluation campaigns, and the EPAC [17] project. The ESTER2 and EPAC data are French broadcast news (prepared speech, plus interviews) of mainly studio quality. The ETAPE data correspond to debates collected from various radio and TV channels (mainly spontaneous speech). The speech data of the ESTER2 and ETAPE train sets, as well as the transcribed data from the EPAC corpus (which amounts to almost 300 hours of signal and almost 4 million running words), were used to train the acoustic models.

In order to assess the performance of our approach for adding new words to a language model, two reference language models are needed: a *baseline* model - to illustrate the performance achieved when the new words are unknown to the system and an *ORACLE* model - trained on a large-vocabulary and on a large data set, to illustrate the maximum performance achieved when the new words are already known and properly trained.

The *ORACLE* language model was trained on various text corpora, using a lexicon of 100k words. The text corpora includes more than 500 million words of newspaper data from 1987 to 2007; several million words from transcriptions of various radio broadcast shows; more than 800 million words from the French Gigaword corpus [18] from 1994 to 2008; plus 300 million words of web data collected in 2011 from various web sources, and thus mainly covering recent years.

The *baseline* language model is trained on the same data as the ORACLE model, but with a few words missing from its vocabulary. We selected a list of 20 nouns, seen between 50 and 100 times in the development sets of ESTER2 and ETAPE. We added to that list their feminine, masculine and plural forms (to avoid the speech recognizer choosing one of those words as replacements based on their similar pronunciation). The final list of 44 words correspond to the English words {evening, place/spot, government, moment, example, problem, power, turn/tower, level, number, group, history, journal, security, meeting, project, year, war, day, report}. These words were removed from the ORACLE's lexicon in order to define the baseline's lexicon, which means that they are 'unknown' words in the baseline language model. In the experiments section they designate the "new words" that will be added to the baseline language model.

Table I describes the sizes of the two language models used as reference in our experiments.

TABLE I. Sizes of the two reference language models

| **Language model** | uni-grams | bi-grams | tri-grams |
|---|---|---|---|
| ORACLE | 97 349 | 43.3M | 80.1M |
| baseline | 97 305 | 42.9M | 79.2M |

The pronunciation variants were extracted from the BDLEX lexicon [19] and from in-house pronunciation lexicons, when available. For the missing words, the pronunciation variants were automatically obtained using JMM-based and CRF-based Grapheme-to-Phoneme converters [20], [21].

The Wikipedia corpus [22] and the GigaWord corpus [18] are used to extract examples of sentences containing the new words. The Wikipedia corpus is also used for finding words similar to the new words. All sentences were tagged with their 'word|PoS-tag' units and with their 'lemma|PoS-tag' units.

### B. Configuration

The SRILM tools [23] were used to create the statistical language models. The TreeTagger software [24] was used to annotate words with lemmas and part-of-speech (PoS) tags.

The Sphinx3 tools [25] were used to train the phonetic acoustic models and to decode the audio signals. The MFCC (Mel Frequency Cepstral Coefficients) acoustic analysis gives 12 MFCC parameters and a logarithmic energy per frame (window of 32 ms, 10 ms shift). The context-dependent phonetic acoustic HMM models were modeled with 64 Gaussian mixtures, and adapted to male and female data.

## III. Methodology

### A. Finding similar words

In the proposed approach, a list of in-vocabulary words is associated to each new word based on the similarity between their neighbor distributions. Two words are considered as similar if they appear in similar contexts.

*1) Computing neighbor distribution of new words:* A data set is needed in order to provide information about the new words. It can be recovered from existing corpora, from the web or be manually composed. The preceding and succeeding neighbors of the new words are extracted from all sentences in order to compute their probability distributions.

Each new word $m$ leads to the determination of the probability distributions $P_k(w|m)$ of all neighbors $w$ found in each position $k$ (related to the new word), with $k = \{\ldots, -3, -2, -1, +1, +2, +3, \ldots\}$.

*2) Computing neighbor distribution of known words:* A different data set is used to represent the 'known' words. Its 2-gram counts define the 2-neighbor distributions (-1,+1), its 3-gram counts define the 4-neighbor distributions (-2,-1,+1,+2), and so on. The preceding and succeeding neighbors of each 'known' word are extracted from the n-gram sequences found in the corresponding counts file in order to compute their probability distributions.

Each 'known' word $x$ leads to the determination of the probability distributions $P_k(w'|x)$ of all neighbors $w'$ found in each position $k$ , with $k = \{\ldots, -3, -2, -1, +1, +2, +3, \ldots\}$.

*3) Comparing neighbor distributions:* The (dis)similarity between the neighbor distributions of a new word *m* and a known word *x* is assessed by the KL divergence [26], on each neighbor position *k*:

$$D_k(x,m) = D_{KL}(P_k(w|x)||P_k(w|m))$$

where $D_{KL}(P||Q) = \sum_w P(w) \cdot log\frac{P(w)}{Q(w)}$.

The KL divergence is computed over the *w* neighbors of the new words in each position *k*. If a word *w* is not present in the known word's neighbors list, its probability is replaced with a default small value $\lambda$ (in our experiments $\lambda = 1e^{-7}$).

The overall (dis)similarity measure between two words is defined as the sum over all neighbor positions *k*:

$$D(x,m) = \sum_k D_k(x,m)$$

The list of most similar words to a new word are those having minimal divergences.

### B. Adding new n-grams to the language model

The following algorithm describes how to define and add ngrams for new words 'nW' that are similar to known words 'kW' (previously saved in the 'similarWords(nW)' list) into a baseline language model 'LM'.

**Algorithm 1** Add new n-grams to a language model

```
1: procedure ADDNGRAMS(nW)
2:     newLM ← LM
3:     newNgrams ← ∅
4: # process the reference ngrams
5:     for each ngram ∈ LM do
6:         for each kW ∈ similarWords(nW) do
7:             if contains(ngram, kW) then
8:                 ngram' ← replace(ngram, kW, nW)
9:                 push(newNgrams, ngram')
10: # choose the new ngrams to add to the newLM
11:    S ← getUniqueSequences(newNgrams)
12:    for each seq ∈ S do
13:        if frequency(seq) = 1 then
14:            prob ← getProbability(seq)
15:        else
16:            P ← getProbabilities(seq)
17:            prob ← medianProbability(P)
18:        push(newLM, "prob seq")
```

For a given new word, this algorithm looks for ngrams related to its similar words and defines new ngrams by replacing the 'similar words' by the 'new word'. Note that replacing the known words with the new words (that they are similar to) might results in multiple ngrams having the same word sequence with different probabilities. In this case, only the median probability is kept for the corresponding word sequence.

## IV. Experiments and Results

### A. Setup for the similar words search

We start the experiments section by presenting the different options used to define the similar words, with a few examples.

*1) New words:* {5,10,20,50} random sentences were extracted for each new word from the Wikipedia and GigaWord corpora and different probability distributions were tested by using {2 neighbors, 4 neighbors, 6 neighbors}.

To give an example of 6 neighbors, let us consider the sentence "les prcipitations sont galement *rparties sur l'* **anne** *avec un total* de 610 millimtres de pluie", where the new word "anne" has: as a -3 neighbor the word 'rparties', as a -2 neighbor the word 'sur', as a +1 neighbor the word 'avec', ..., and as a +3 neighbor the word 'total'.

*2) Known words:* The 'known' words belong to the Wikipedia corpus and to the baseline lexicon.

From the 4-gram sequence "cdent leur place ": the word cdent has a +1 neighbor 'leur', a +2 neighbor 'place' and a +3 neighbor ''; the word leur has a -1 neighbor 'cdent', a +1 neighbor 'place' and a +2 neighbor ''; the word place has a -2 neighbor 'cdent', a -1 neighbor 'leur' and a +1 neighbor ''; the word  has a -3 neighbor 'cdent', a -2 neighbor 'leur' and a -1 neighbor 'place'.

*3) Known words associated to new words:* Different lists of similar words are obtained when using either word-based sentences, 'word|PoStag'-based sentences or 'lemma|PoStag'-based sentences.

Here is an example of 10 similar words (translated from French to English for better comprehension) obtained for the new word "journal", based on 10 examples of sentences, 4-gram Wikipedia counts and 6 neighbor distributions:

- based on word sentences : name, first, title, game, book, even, world, magazine, novel, second
- based on word|PoStag sentences : magazine, name, title, game, book, world, service, text, program, network
- based on lemma|PoStag sentences : chronicle, title, magazine, name, series, book, version, game, program, press.

Note that words can have different meanings in different contexts. Also, the 'lemma|PoStag' sentences can not be used when adding feminine, masculine and plural forms of words (since all words are reduced to root form).

The similar words obtained on the "word|PoStag" sentences and on the 6 neighbors probability distributions are the only ones considered in the experiments reported next.

### B. New language models

Two different settings were evaluated in our experiments: adding only the uni-grams or all the n-grams (unigrams, bigrams and trigrams) computed for the 44 new words to the baseline language model.

Several lists of similar words have been evaluated when using {5, 10, 20, 50} examples of sentences for each new word. Only 10 word matches are considered for each new word.

TABLE II. NUMBER OF BI-GRAMS AND TRI-GRAMS OF THE NEW 'BASELINE+ADDEDNGRAMS' LANGUAGE MODELS

| | # examples of sentences | | | |
|---|---|---|---|---|
| | 5 | 10 | 20 | 50 |
| #bi-grams | 44.65M | 44.63M | 44.75M | 44.79M |
| #tri-grams | 89.77M | 89.27M | 90.45M | 90.79M |

TABLE III. REFERENCE STATISTICS OBTAINED ON BOTH REFERENCE LANGUAGE MODELS

| **Language model** | **globalWER** | **new words correctRec.** |
|---|---|---|
| ORACLE | 24.79 | 84.91 |
| baseline | 26.79 | 0.00 |

Based on these 4 lists of similar words we created 4 different new language models with added uni-grams (baseline+1grams) and 4 different new language models with added n-grams (baseline+Ngrams).

The 'baseline+1grams' LMs have only an increased number of uni-grams (from 97305 to 97349) compared to the baseline LM. The number of bi-grams and tri-grams associated with the 'baseline+Ngrams' LMs are presented in table II. They add between 1.7 to 1.9 million bi-grams and between 10.6 to 11.6 million tri-grams to the baseline LM.

### C. Decoding performance of different LMs

The language models are evaluated over the ESTER2 development data set, in which the set of 44 words have a total occurrence frequency of 1.33%.

Table III displays the word-error-rates (*WER*) and the percentage of new words that are correctly recognized with both reference language models. The difference of 2% in the WER performance is due to the 44 words that are unknown in the baseline language model.

Table IV and V present the WER and the percentage of new words that are correctly recognized with the new language models, when using various amounts of example sentences per new word (5, 10, 20, 50). Adding only 1-grams for new words to the LM ('baseline+1grams') hardly improves the WER, and correctly recognizes only 25% of the new words. However, adding n-grams for new words to the LM ('baseline+Ngrams') provides 1.30% absolute WER improvement over the baseline model and is only 0.70% worse than the ORACLE model. Moreover, it correctly recognizes up to 65% of the new words. Good results can be achieved with 5, 10 examples of sentences per each new word (using more examples provides no real improvement).

These results show that our similarity approach and our method to add new n-grams to a language model are efficient.

## V. CONCLUSIONS

This paper proposed a new approach to directly define and add into a word-based language model n-gram entries for new words, based on the similarity of the new words with in-vocabulary words.

Our approach is based on similar neighbor distributions (two words are considered as similar if they appear in similar contexts) and it requires very little data related to the new words (5 sentences for each new word are sufficient to achieve good results). The n-gram parameter values associated to the similar words are then used to define the n-gram parameter values of the new words.

TABLE IV. WER OF THE NEW 'BASELINE+ADDED1GRAMS' AND 'BASELINE+ADDEDNGRAMS' LMS ON THE ESTER2 DEVELOPMENT SET

| | # examples of sentences | | | |
|---|---|---|---|---|
| | 5 | 10 | 20 | 50 |
| baseline+added1grams | 26.45 | 26.44 | 26.40 | 26.42 |
| baseline+addedNgrams | 25.68 | 25.51 | 25.51 | 25.57 |

TABLE V. PERCENTAGE OF NEW WORDS THAT ARE CORRECTLY RECOGNIZED WITH THE NEW 'BASELINE+ADDED1GRAMS' AND 'BASELINE+ADDEDNGRAMS' LMS ON THE ESTER2 DEVELOPMENT SET

| | # examples of sentences | | | |
|---|---|---|---|---|
| | 5 | 10 | 20 | 50 |
| baseline+added1grams | 29.81 | 20.00 | 22.18 | 20.36 |
| baseline+addedNgrams | 60.54 | 61.81 | 64.90 | 62.76 |

Adding only 1-grams for new words hardly improves the performance. However, adding n-grams (i.e., 1-grams, 2-grams and 3-grams) for new words provides results close to the ORACLE's performance. The results shows that our similarity approach and our method to add new n-grams into a language model are efficient.

Future work will investigate further the setups for finding similar words. The n-grams of new words will also be filtered in order to diminish the size of new language models. The impact of other parameters will also be evaluated (e.g. the number of similar words considered for each new word).

## ACKNOWLEDGEMENTS

The work presented in this article is part of the RAPSODIE project, and has received support from the "Conseil Régional de Lorraine" and from the "Région Lorraine" (FEDER) (http://erocca.com/rapsodie).

## REFERENCES

[1] S. Young, "A review of large-vocabulary continuous-speech," *Signal Processing Magazine, IEEE*, vol. 13, no. 5, p. 45, 1996.

[2] P. C. Woodland, S. E. Johnson, P. Jourlin, and K. S. Jones, "Effects of out of vocabulary words in spoken document retrieval," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2000, pp. 372–374.

[3] H. Ketabdar, M. Hannemann, and H. Hermansky, "Detection of out-of-vocabulary words in posterior based ASR," in *Proceedings of Interspeech*, 2007, pp. 1757–1760.

[4] J. R. Bellegarda, "Statistical language model adaptation: review and perspectives," *Speech Communication*, vol. 42, pp. 93–108, 2004.

[5] F. Keller and M. Lapata, "Using the web to obtain frequencies for unseen bigrams," *Computational Linguistics*, vol. 29, no. 3, pp. 459–484, 2003.

[6] B. Suhm and A. Waibel, "Towards better language models for spontaneous speech," in *The 3rd International Conference on Spoken Language Processing (ICSLP)*. ISCA, 1994.

[7] A. Asadi, R. Schwartz, and J. Makhoul, "Automatic modeling for adding new words to a large-vocabulary continuous speech recognition system," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 1991, pp. 305–308.

[8] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. Della Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, pp. 467–479, 1992.

[9] A. Pražák, P. Ircing, and L. Müller, "Language model adaptation using different class-based models," in *Proceedings of SPECOM*, 2007, pp. 449–454.

[10] W. Naptali, M. Tsuchiya, and S. Nakagawa, "Class-based n-gram language model for new words using out-of-vocabulary to in-vocabulary similarity," *IEICE Transactions on Information and Systems*, vol. E95-D, no. 9, pp. 2308–2317, 2012.

[11] I. Zitouni, "Backoff hierarchical class n-gram language models: effectiveness to model unseen events in speech recognition," *Computer Speech & Language*, vol. 21, no. 1, pp. 88–104, 2007.

[12] C. Martins, A. J. S. Teixeira, and J. P. Neto, "Automatic estimation of language model parameters for unseen words using morpho-syntactic contextual information," in *Proceedings of Interspeech*, 2008, pp. 1602–1605.

[13] G. Lecorv, G. Gravier, and P. Sbillot, "Automatically finding semantically consistent n-grams to add new words in LVCSR systems," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 4676–4679.

[14] I. Dagan, L. Lee, and F. C. N. Pereira, "Similarity-based models of word cooccurrence probabilities," in *Machine Learning*, vol. 34, no. 1-3, 1999, pp. 43–69.

[15] S. Galliano, G. Gravier, and L. Chaubard, "The ESTER 2 evaluation campaign for rich transcription of French broadcasts," in *Proceedings of Interspeech*, 2009.

[16] G. Gravier, G. Adda, N. Paulson, M. Carr, A. Giraudel, and O. Galibert, "The ETAPE corpus for the evaluation of speech-based TV content processing in the French language," in *Proceedings of the International Conference on Language Resources, Evaluation and Corpora*, 2012.

[17] Y. Estve, T. Bazillon, J.-Y. Antoine, F. Bchet, and J. Farinas, "The EPAC corpus: Manual and automatic annotations of conversational speech in French broadcast news," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, 2010.

[18] A. Mendona, D. Graff, and D. DiPersio, "French Gigaword third edition," in *Proceedings of the Linguistic Data Consortium*, 2011.

[19] M. de Calms and G. Prennou, "BDLEX: a lexicon for spoken and written French," in *Language Resources and Evaluation*, 1998, pp. 1129–1136.

[20] I. Illina, D. Fohr, and D. Jouvet, "Grapheme-to-phoneme conversion using conditional random fields," in *Proceedings of Interspeech*, 2011, pp. 2313–2316.

[21] D. Jouvet, D. Fohr, and I. Illina, "Evaluating grapheme-to-phoneme converters in automatic speech recognition context," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4821–4824.

[22] F. Sajous, "Corpus WikipdiaFR2008," "http://redac.univ-tlse2.fr/corpus/wikipedia.html".

[23] A. Stolcke, "SRILM an extensible language modeling toolkit," in *Conference on Spoken Language Processing*, 2002.

[24] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *Proceedings of the International Conference on New Methods in Language Processing*, 1994, pp. 44–49.

[25] P. Placeway, S. Chen, M. Eskenazi, U. Jain, V. Parikh, B. Raj, M. Ravishankar, R. Rosenfeld, K. Seymore, M. Siegler, R. Stern, and E. Thayer, "The 1996 Hub-4 Sphinx-3 system," in *DARPA Speech Recognition Workshop*, 1996.

[26] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.